[illegible]

Final Report

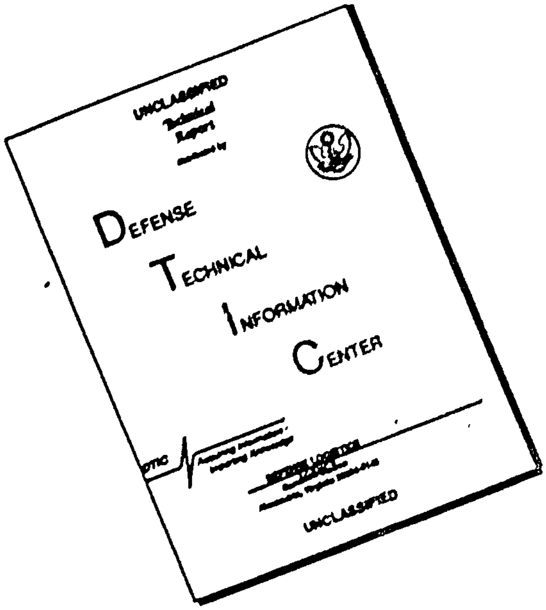
LOW ENERGY PHOTOELECTRON [illegible]
[illegible] CALCULATIONS

Vol. III: Program Description

[illegible]

UNION CARBIDE CORPORATION

DISCLAIMER NOTICE



UNCLASSIFIED
Technical
Report
[illegible]

DEFENSE
TECHNICAL
INFORMATION
CENTER

DTIC [illegible]

[illegible]

UNCLASSIFIED

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNION CARBIDE CORPORATION

DEFENSE AND SPACE SYSTEMS DEPARTMENT

UCC/DSSD - 299

Final Report

LOW ENERGY PHOTOELECTRIC CROSS SECTION CALCULATIONS

Vol. II: Program Description

A. Glick and H. Brysk

July 31, 1967

Work Performed under Contract No. DA-49-146-XZ-511

Each transmittal of this document outside
the agencies of the U.S. Government must
have prior approval of the Director, Defense
Atomic Support Agency, Washington, D.C. 20305.

Space Sciences and Engineering Laboratory
5 New Street
White Plains, New York 10601

10

FOREWORD

This report was prepared by the Union Carbide Corporation, Defense and Space Systems Department, White Plains, New York, under Contract DA-49-146-XZ-511, Project 5710, funded by the Defense Atomic Support Agency (DASA). Inclusive dates of research were 1 June 1966 to 30 June 1967. The report was submitted 31 August 1967 by the AFWL Technical Monitor, Captain Guy Spitale (WLRP).

The report has been divided into two volumes for convenience. Volume I presents the theoretical analysis and the discussion of results. Volume II is a detailed description of the program.

The project was initiated and formulated by Dr. C. D. Zerby. The theoretical derivation was completed by Dr. H. Brysk, who also planned and analyzed the calculations. Programming support was supplied by Mr. A. Glick in writing the program and by Mr. E. C. Imperatore in resolving the systems problems of transcribing the tapes containing the Los Alamos Scientific Laboratory (LASL) self-consistent-field data to the UCC and the AFWL operating systems. The program is written in FORTRAN IV and is operational on the CDC 6600 computer at AFWL. We thank Dr. J. T. Waber (of LASL) who supplied us the output of his self-consistent-field program on tape.

This report has been reviewed and is approved.

ABSTRACT

A computer program was developed for the calculation of photoelectric cross sections, including angular distributions, using as input the results of a relativistic Dirac-Slater self-consistent-field program. The program was used to calculate the aluminum cross sections over the range from 1 to 150 keV and uranium cross sections at four energies within that range, and the results were correlated with pre-existent experimental and theoretical data.

Volume I presents the theoretical analysis and the discussion of results. Volume II is a detailed description of the program.

This page is intentionally left blank.

TABLE OF CONTENTS

I. INTRODUCTION

This volume is a self-contained description of a program (PELEC) for the computation of photoelectric cross sections. The companion volume (UCC/DSSD - 299, Volume I) develops the theory used and discusses the results obtained with the program.

Chapter II contains the operating instructions. It describes the input cards and self-consistent-field data tape required. The utilization of a test option to ascertain the optimum input parameters in a series of runs is indicated.

Chapter III presents sample output.

Chapter IV exhibits the program itself. For each routine, its purpose and the method of achieving it are stated. The routines it calls and those it is called by are given, as well as the Common blocks it uses and the calling sequence (if any). The variables are defined. A schematic flow chart is provided. Finally, the routine is listed.

This page is intentionally left blank.

II. OPERATING INSTRUCTIONS

This chapter gives all the operational details needed in order to run the photoelectric program. The input variables and format are listed, along with the program diagnostics. Tape unit assignments are given and the test option is described.

Input Variables

A complete set of data consists of a single data card. The variables described below are read in on a format (9I5, 2E15.8):

5	10	15	20	25	30	35	40	45	60	75
JM	KMAX	LM	NTAPE	NEDGE	IA	IB	IZ	LØØP	QV	SAVE

Upon completion of a problem, the program recycles, reading in the next data card. Termination of the program is obtained by setting the flag value NEDGE = -1.

Name	Dimensions	Mode	Meaning
JM		I	Maximum order of Legendre coefficient
KMAX		I	Maximum κ for electron
LM		I	Maximum ℓ for photon
NTAPE		I	The logical tape unit assigned for tape containing the self-consistent-field data
NEDGE		I	= -1: terminates Program = 0: normal run > 0: sets photon energy to NEDGE th binding energy
IA		I	= 0: normal case > 0: calculation commences after IA shells
IB		I	= 0: calculation to include outermost shell > 0: calculation cuts off after IB shells
IZ		I	Atomic number
LØØP		I	= 0: normal case = 1: photon angular momentum reduction
QV		R	Photon energy in keV
SAVE		R	Total cross section accumulation from previous run; = 0 ordinarily.

Input Testing

The program sifts the input data to insure that certain criteria are not violated. If any difficulty is observed, a violation signal is printed and the run terminated. Listed below in abbreviated form are the criteria that must be satisfied and that the program tests for:

MIN (JM, KMAX, LM, IA, IB) $\geq$ 0

$2 < IZ < 102$

$QV > 0$

LØØP = 0 or 1

NEDGE $\leq$ JX

IA $<$ JX

IB $\leq$ JX

NK, NKP $\leq$ 200

In addition to the above the program examines the following variables and alters them if necessary:

JM > 24, program sets JM = 24, prints this fact and proceeds.

KMAX > 12, program sets KMAX = 12, prints this fact and proceeds.

LM > 12, program sets LM = 12, prints this fact and proceeds.

This is to insure that the dimension size of the variable of the program as written for the IBM 7094 is not exceeded.

Program Options

The program has two option procedures controlled by LØØP and by the combination of values of IA, IB, and SAVE.

By setting LØØP = 1, we can reduce the original angular momentum quantity LM (max ℓ for photon) in unit increments (until a minimum of one is reached). The selection rules may then reduce the range limits KM and JM. The summing process is repeated to recalculate the differential and total cross sections.

By altering the input values IA and IB we can segment a run. The program proceeds through IB electron shells rather than the totality of shells (JX). The total cross section up to that point can be fed back in as part of the input (SAVE) when the user desires to continue the run. With IA greater than zero, the program starts after the first IA electron shells. This process is useful when long computer running time is not available.

System Information

The program as run at the New York Regional Computer Center utilizes the following tapes:

Logical tape unit 5	Read
Logical tape unit 6	Write
Logical tape unit 1	Self-consistent-field data

This page is intentionally left blank.

III. SAMPLE OUTPUT

The output from an actual run is given. The complete output contains the differential cross section for each subshell in succession. Only one subshell is reproduced here, for economy.

```
INPUT DATA FOR PHOTOELECTRIC CALCULATION

INPUT CARD READS

JM  KMAX  LM  NTAPE  NEDGE  IA  IB  IZ  LOOP        OV              SAVE
 11    6    4    1     0     0    0  13     0 0.10000000E 02 0.
```

```
INPUT DATA FOR PHOTOELECTRIC CALCULATION

          13 = NUCLEAR CHARGE
      10.000 = PHOTON ENERGY

           6 = MAX KAPPA FOR ELECTRON
           4 = MAX L FOR PHOTON
          11 = MAX J (LEGENDRE COEFF.)




          TAPE POSITIONED PROPERLY.




                   AL       = ELEMENT
                         13 = ATOMIC NUMBER
                          6 = NUMBER OF ELECTRON SHELLS
             0.19449370E 00 = SCREENING FACTOR OF OUTERMOST BOUND ELECTRON
                        257 = RADIAL GRID UP TO X =  1.0
                       1281 = RADIAL GRID UP TO X =  65.0
                       1436 = TOTAL RADIAL GRID
             0.82222162E 04 = OUTERMOST RADIAL VALUE
```

```
                    1S1/2 = SHELL
         0.27269274E 03 = INTEGRATION CUT-OFF
                     1327 = NUMBER OF WAVE FUNCTION GRID POINTS
                        5 = MAX KAPPA FOR THIS SHELL
                       25 = NUMBER OF MATRIX ELEMENTS FOR THIS SHELL

LENGTH UNITS ARE HBAR / MC ( 1 BOHR RADIUS = 137 )

INTEGRATION STEP SIZE IS  0.0078125     UP TO    1.000

INTEGRATION STEP SIZE IS  0.1250000     UP TO   65.000

INTEGRATION STEP SIZE IS  1.0000000     UP TO  272.693
```

```
             11 = MAX J (LEGENDRE COEFF.)
              5 = MAX KAPPA FOR ELECTRON
              4 = MAX L FOR PHOTON


        LEGENDRE COEFFICIENTS OF CROSS SECTION

    J           D(J)

    0       0.84832748E 02
    1       0.34841149E 02
    2      -0.76300052E 02
    3      -0.33218958E 02
    4      -0.82906764E 01
    5      -0.15982244E 01
    6      -0.22848516E 00
    7      -0.24635064E-01
    8      -0.20869956E-02
    9      -0.11737233E-03
```

```
                 ELEMENT        ATOMIC NUMBER            SHELL

                    AL               13                  1S1/2

BINDING ENERGY                  PHOTON ENERGY        ELECTRON KINETIC ENERGY

 1.5499309 KEV                 10.0000000 KEV               8.4500691 KEV

          UNPOLARIZED CROSS SECTION (BARNS/STERADIAN)

THETA            COS THETA                CROSS SECTION              ANG. DIST.

   0             1.0000000                0.10482765E-01             0.0000778
   2             0.9993908                0.31837447E 00             0.0023632
   4             0.9975640                0.12387132E 01             0.0091947
   6             0.9945219                0.27615944E 01             0.0204968
   8             0.9902681                0.48706563E 01             0.0361540
  10             0.9848078                0.75433565E 01             0.0559929
  12             0.9781476                0.10751303E 02             0.0798049
  14             0.9702957                0.14460679E 02             0.1073389
  16             0.9612617                0.18632747E 02             0.1383074
  18             0.9510565                0.23224424E 02             0.1723906
  20             0.9396926                0.28188894E 02             0.2092409
  22             0.9271839                0.33476281E 02             0.2484882
  24             0.9135455                0.39034332E 02             0.2897446
  26             0.8987940                0.44809127E 02             0.3326098
  28             0.8829476                0.50745737E 02             0.3766766
  30             0.8660254                0.56789170E 02             0.4215355
  32             0.8480481                0.62884551E 02             0.4667804
  34             0.8290376                0.68978249E 02             0.5120127
  36             0.8090170                0.75018242E 02             0.5568465
  38             0.7880108                0.80954696E 02             0.6009117
  40             0.7660444                0.86740461E 02             0.6438584
  42             0.7431448                0.92331495E 02             0.6853596
  44             0.7193398                0.97687210E 02             0.7251140
  46             0.6946584                0.10277077E 03             0.7628484
  48             0.6691306                0.10754931E 03             0.7983186
  50             0.6427876                0.11199406E 03             0.8313111
  52             0.6156615                0.11608047E 03             0.8616438
  54             0.5877853                0.11978819E 03             0.8891655
  56             0.5591929                0.12310107E 03             0.9137564
  58             0.5299193                0.12600702E 03             0.9353266
  60             0.5000000                0.12849794E 03             0.9538164
  62             0.4694716                0.13056947E 03             0.9691930
  64             0.4383712                0.13222094E 03             0.9814509
  66             0.4067366                0.13345459E 03             0.9906087
  68             0.3746066                0.13427632E 03             0.9967063
  70             0.3420202                0.13469443E 03             0.9998118
  72             0.3090170                0.13471978E 03             1.0000000
  74             0.2756374                0.13436550E 03             0.9973702
  76             0.2419219                0.13364660E 03             0.9920340
  78             0.2079117                0.13257975E 03             0.9841149
  80             0.1736482                0.13118296E 03             0.9737468
  82             0.1391731                0.12947536E 03             0.9610716
  84             0.1045285                0.12747693E 03             0.9462376
 86          0.0697565          0.12520828E 03          0.9293979
 88          0.0348995          0.12269040E 03          0.9107082
 90          0.0000000          0.11994453E 03          0.8903260
 92         -0.0348995          0.11699190E 03          0.8684091
 94         -0.0697564          0.11385362E 03          0.8451144
 96         -0.1045284          0.11055058E 03          0.8205965
 98         -0.1391731          0.10710323E 03          0.7950075
100         -0.1736482          0.10353156E 03          0.7684956
102         -0.2079117          0.99854979E 02          0.7412050
104         -0.2419219          0.96092227E 02          0.7132748
106         -0.2756373          0.92261351E 02          0.6848389
108         -0.3090170          0.88379639E 02          0.6560257
110         -0.3420201          0.84463592E 02          0.6269576
112         -0.3746066          0.80528891E 02          0.5977510
114         -0.4067366          0.76590403E 02          0.5685164
116         -0.4383711          0.72662151E 02          0.5393577
118         -0.4694715          0.68757333E 02          0.5103729
120         -0.5000000          0.64888321E 02          0.4816540
122         -0.5299192          0.61066679E 02          0.4532866
124         -0.5591929          0.57303176E 02          0.4253509
126         -0.5877852          0.53607805E 02          0.3979208
128         -0.6156615          0.49989820E 02          0.3710652
130         -0.6427876          0.46457748E 02          0.3448472
132         -0.6691306          0.43019425E 02          0.3193252
134         -0.6946583          0.39682028E 02          0.2945523
136         -0.7193398          0.36452100E 02          0.2705772
138         -0.7431448          0.33335594E 02          0.2474439
140         -0.7660444          0.30337900E 02          0.2251926
142         -0.7880107          0.27463871E 02          0.2038592
144         -0.8090170          0.24717871E 02          0.1834762
146         -0.8290376          0.22103603E 02          0.1640724
148         -0.8480481          0.19625136E 02          0.1456737
150         -0.8660254          0.17284947E 02          0.1283030
152         -0.8829476          0.15085945E 02          0.1119802
154         -0.8987940          0.13030508E 02          0.0967230
156         -0.9135454          0.11120710E 02          0.0825470
158         -0.9271838          0.93583400E 01          0.0694652
160         -0.9396926          0.77449520E 01          0.0574893
162         -0.9510565          0.62818686E 01          0.0466291
164         -0.9612617          0.49702084E 01          0.0368929
166         -0.9702957          0.38109178E 01          0.0282877
168         -0.9781476          0.28047778E 01          0.0208193
170         -0.9848078          0.19524343E 01          0.0144926
172         -0.9902681          0.12544079E 01          0.0093112
174         -0.9945219          0.71109624E 00          0.0052783
176         -0.9975640          0.32280832E 00          0.0023961
178         -0.9993908          0.89753443E-01          0.0006662
180         -1.0000000          0.12053592E-01          0.0000895


          INTEGRATED CROSS SECTION =  0.10660397E 04 BARNS
```

```
ELEMENT                                    ATOMIC NUMBER

  AL                                            13

PHOTON ENERGY =    10.0000000  KEV

         TOTAL CROSS SECTION =  0.11461591E 04  BARNS
```

IV. THE PROGRAM

The program is written in FORTRAN IV. The versions used on the IBM 7094 and the CDC 6600 are identical except for control cards. The data tapes contain the same information in binary form, but are not compatible between the two machines.

All the Common blocks appear in the main routine (PELEC). The definition of the Common variables is given first. In the subroutines, Common blocks used are quoted. The unlabelled Common is the same wherever it appears.

Definition of Variables in COMMON

Unlabelled Common: Length 107_8

Name	Dimensions	Mode	Meaning
PI		R	π
HALFPI		R	$\pi/2$
FØURPI		R	4π
RAD		R	$\pi/180$
SQ2		R	$2^{-1/2}$
Q		R	Photon energy (in mc^2 units)
ZA		R	Atomic number
ZAZA		R	ZA * ZA
EFN		R	Free electron energy -1 (in mc^2 units)
EGN		R	Free electron energy +1 (in mc^2 units)
V		R	Potential; screening factor/radius
CG	30	R	$-\varkappa - \gamma$ if radius < 1; $-\varkappa$ if radius > 1
GAM	30	R	$\sqrt{\varkappa^2 - ZAZA}$

/BESSEL/ Common: Length 420_8

Name	Dimensions	Mode	Meaning
FL	15	R	Numerical factors used in the construction of the spherical Bessel function.
PC	15	R	
ØF	15 x 15	R	
M1		I	Largest order of Bessel function needed
M2		I	M1 + 1
B	15	R	Spherical Bessel function

/DFUNC/ Common: Length 1047_8

Name	Dimensions	Mode	Meaning
F	30	R	"Small" component of free-electron wavefunction
G	30	R	"Large" component of free-electron wavefunction
DF	30	R	Derivative of F
DG	30	R	Derivative of G
DFK	200	R	Integrand for matrix elements $K_\ell(\kappa\kappa')$
DFKP	200	R	Integrand for matrix elements $K_\ell(\kappa'\kappa)$
CF	30	R	$\kappa - \gamma$ if radius < 1; κ if radius > 1
H		R	Integration step size

/FAC/Common: Length 324_8

Name	Dimensions	Mode	Meaning
FACT	67	R	Numerical factors used in the calculation of the Clebsch-Gordan coefficients.
RTFACT	95	R	
RØØT	50	R	

/FIDØ/ Common: Length 746_8

Name	Dimensions	Mode	Meaning
FI	30 x 15	R	$\phi(\kappa,\lambda)$
D	30	R	Legendre coefficient of cross section
JMP		I	Max. order of Legendre coefficient + 1
NAME		ALFA	Element
SHELL		ALFA	Electron shell
QV		R	Photon energy (in keV units)
EB		R	Binding energy of shell (in keV units)
IZ		I	Atomic number

/KUT/ Common: Length 24_8

Name	Dimensions	Mode	Meaning
RK1	4	R	Numerical coefficients used for the Runge-Kutta integration (Gill Form)
RK2	4	R	
RK3	4	R	
RK4	4	R	
K4	4	I	

/LIMIT/ Common: Length 13_8

Name	Dimensions	Mode	Meaning
JM		I	Max order of Legendre coefficient
LM		I	Max ℓ for photon
KM		I	Max κ for free electron
K2M		I	2 * KM; number of free electron states
IEND		I	Flag for zero electron kinetic energy state
NEW		I	Flag to save repetition of calculation when radius is not advanced
NK		I	Number of matrix elements $K_\ell(\kappa\kappa')$
NKP		I	Number of matrix elements $K_\ell(\kappa'\kappa)$
JKB		I	Twice $j_{\kappa'}$
LMKB		I	$\ell_{-\kappa'}$
NTAB		I	Radial index in bound state tabulation

/MAT/ Common: Length 1535_8

Name	Dimensions	Mode	Meaning
SF	30	R	Integration storage variable for "small" component of free electron wavefunction
SG	30	R	Ingegration storage variable for "large" component of free electron wavefunction
FK	200	R	Matrix elements $K_\ell(\kappa\kappa')$
FKP	200	R	Matrix elements $K_\ell(\kappa'\kappa)$

Name	Dimensions	Mode	Meaning
SFK	200	R	Integration storage variable for matrix elements $K_\ell(\kappa\kappa')$
SFK2	200	R	Integration storage variable for matrix elements $K_\ell(\kappa'\kappa)$
RCUT		R	Cut-off radius for integration

/QUANT/ Common: Length 322_8

Name	Dimensions	Mode	Meaning
LK	30	I	ℓ_κ
LMK	30	I	$\ell_{-\kappa}$
JK	30	I	Twice j_κ
FKAP	30	R	κ
SN	30	R	Sign of κ
SI	30	R	Sine of phase shift
CR	30	R	Cosine of phase shift

/ONWARD/ Common: Length 4_8

Name	Dimensions	Mode	Meaning
RX		R	Radial variable beyond radial cutoff on uniform grid tabulation
SCX		R	Interpolated value of screening factors
GBX		R	Interpolated value of "large" component of wavefunction
FBX		R	Interpolated value of "small" component of wavefunction

/TAPES/ Common: Length 13562_8

Name	Dimension	Mode	Meaning
X	1500	R	Radial value
SCF	1500	R	Screening factor
FB	1500	R	"Small" component of bound state wavefunction

Name	Dimensions	Mode	Meaning
GB	1500	R	"Large" component of bound state wavefunction
GAMB		R	$\sqrt{\kappa^2 - ZAZA}$ for bound state
SCREEN		R	Normalization factor for least bound electron

/TRANS/ Common: Length 1700_8

Name	Dimensions	Mode	Meaning
HF	30 x 15	R	$H(\kappa,\mu')$ for $\mu' > 0$
HFM	30 x 15	R	$H(\kappa,\mu')$ for $\mu' < 0$
JNG	30	I	$(-\mu')$ max + 1/2
JPS	30	I	(μ') max + 1/2 provided $\mu' > 0$ permitted; -1 otherwise

/VECT/ Common: Length 1441_8

Name	Dimensions	Mode	Meaning
KF	200	I	Index for κ values for $K_\ell(\kappa'\kappa)$
KG	200	I	Index for κ values for $K_\ell(\kappa\ \kappa')$
LBES	200	I	Photon angular momentum + 1 for $K_\ell(\kappa'\ \kappa)$
LBS	200	I	Photon angular momentum + 1 for $K_\ell(\kappa\ \kappa')$
LKB		I	$\ell_{\kappa'}$

ROUTINE PELEC

Purpose: This is the main program. It serves as the control section, reading the input cards and the tape containing the self-consistent-field data and setting up the calculations. Nearly all the actual computation is done in subroutines.

Method: Numerical constants are computed and stored, the input read and checked (and diagnostics printed, if needed), then printed with interpretation; the tape is scanned for the element required and the data read from it and interpreted. For each subshell, SINDEX is called to set up the matrix elements and their quantum numbers, RADINT to perform the radial integrations. LEGEND is called to perform the angular momentum sums, ANGLE to produce the differential cross section (headings are supplied for the tables printed in the subroutines); if the matrix reduction option is called for, the cutoff values of the quantum numbers are reduced and the program loops back through the angular sums. The total cross section is accumulated in PELEC and printed out. The program recycles to read the next input card and perform the next case, until a flag on the input card signals termination.

Subroutines called: SINDEX, RADINT, LEGEND, ANGLE, HUM

Variables in unlabelled Common: PI, HALFPI, FØURPI, RAD, SQ2, Q, ZA, ZAZA,
EFN, EGN, V, CG, GAM

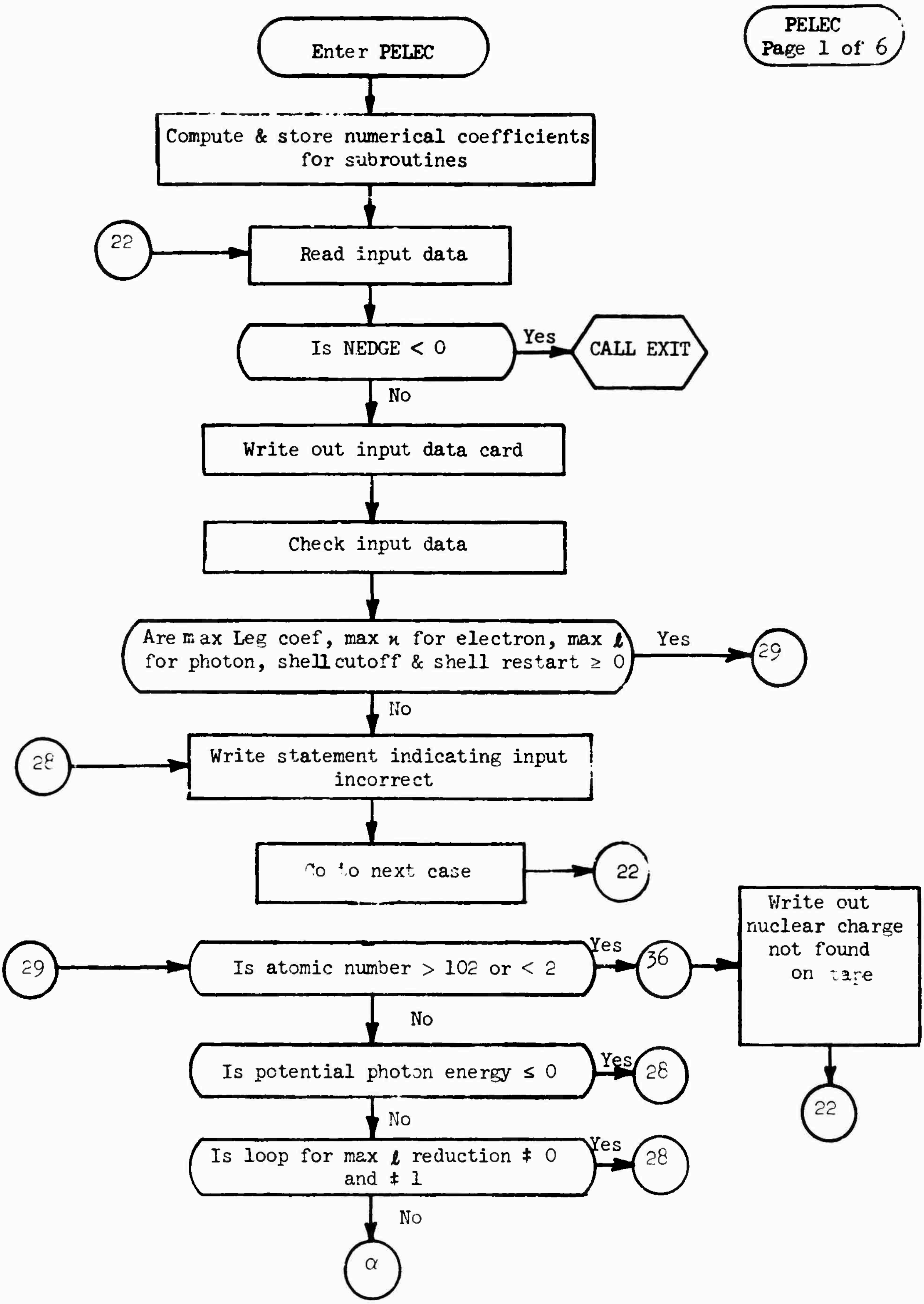
Labelled Common: BESSEL, DFUNC, FAC, FIDØ, KUT, LIMIT, MAT, QUANT, ØNWARD,
TAPES, TRANS, VECT

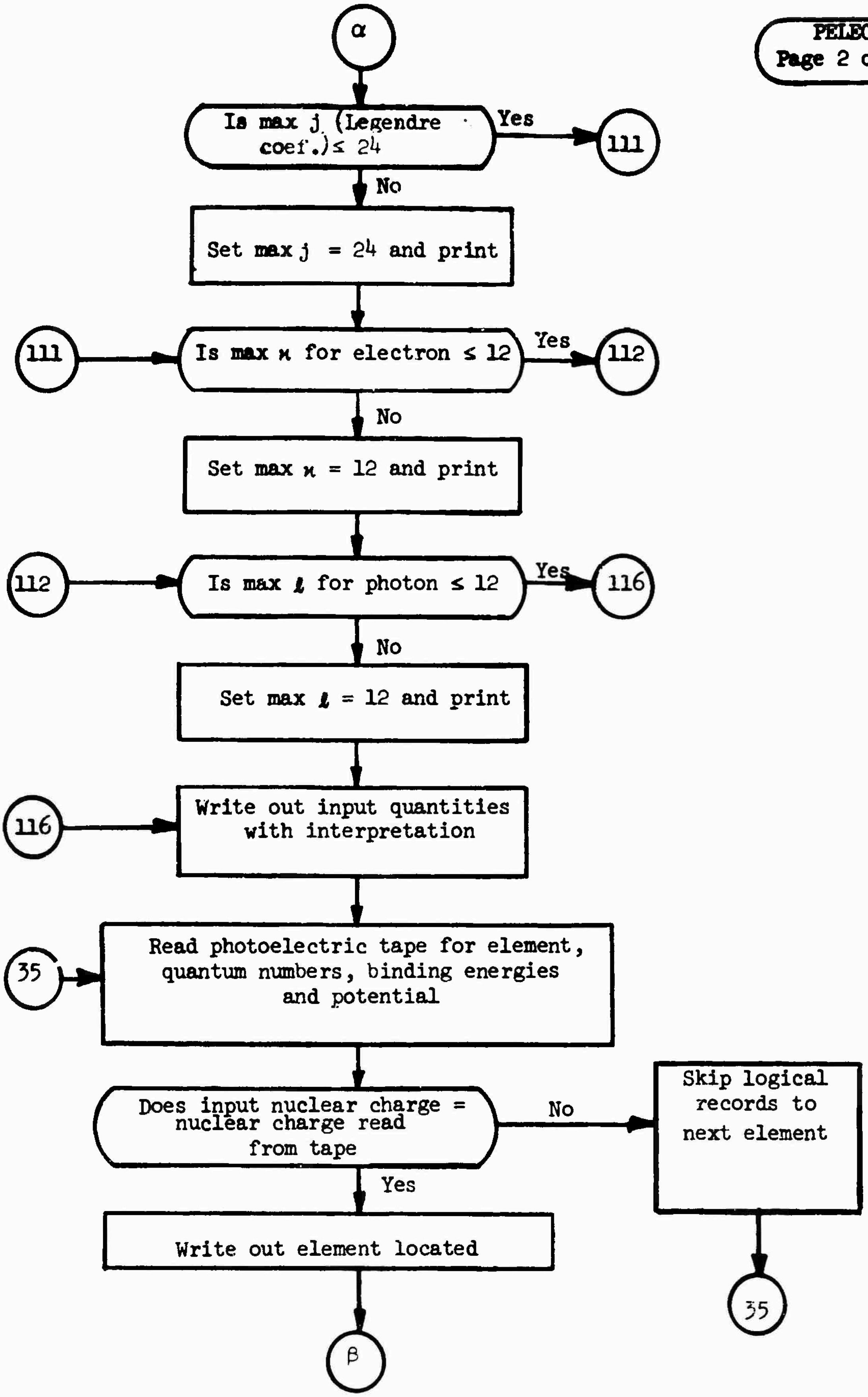
Local Variables:

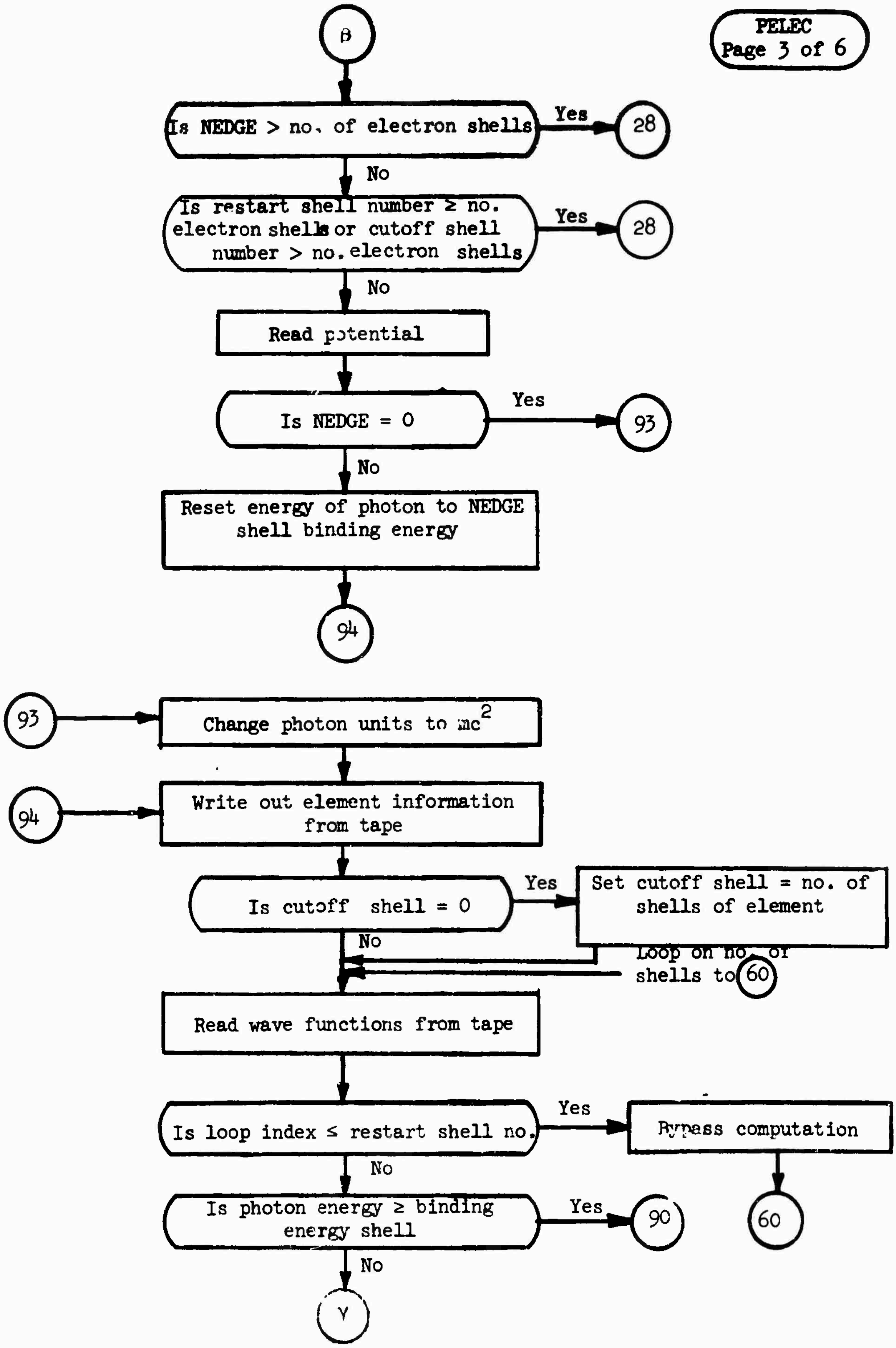
Name	Dimension	Mode	Meaning
KMAX		I	Input max κ for free electron
NTAPE		I	The logical tape unit assigned for tape containing the self-consistent-field data
NEDGE		I	= -1 terminates program = 0 normal run = integer sets photon energy to NEDGEth binding energy
IA		I	= 0 normal case > 0 calculation commences after IA shells
IB		I	= 0 calculation includes outermost shell > 0 cuts off calculation after IB shells
LØØP		I	= 0 normal case = 1 for photon angular momentum reduction
QV		R	Input photon energy in keV
SAVE		R	Input total cross section; accumulation from previous run; = 0 otherwise
IZ		I	Input atomic number
NGRID		I	Number of grid points in table
ISIXFI		I	Number of grid points for radial value of 65 (1/2 Bohr unit)
IØNE		I	Number of grid points for radial value of one
RSIXFI		R	Radial value of 65, read from tape
RØNE		R	Radial value of one, read from tape
ZTRY		R	Atomic number read from tape
JX		I	Number of electron shells
XN	36	ALFA	Shell identification
XL	36	R	$l_{\kappa'}$
XJ	36	R	$j_{\kappa'}$
XZ	36	R	Shell occupancy

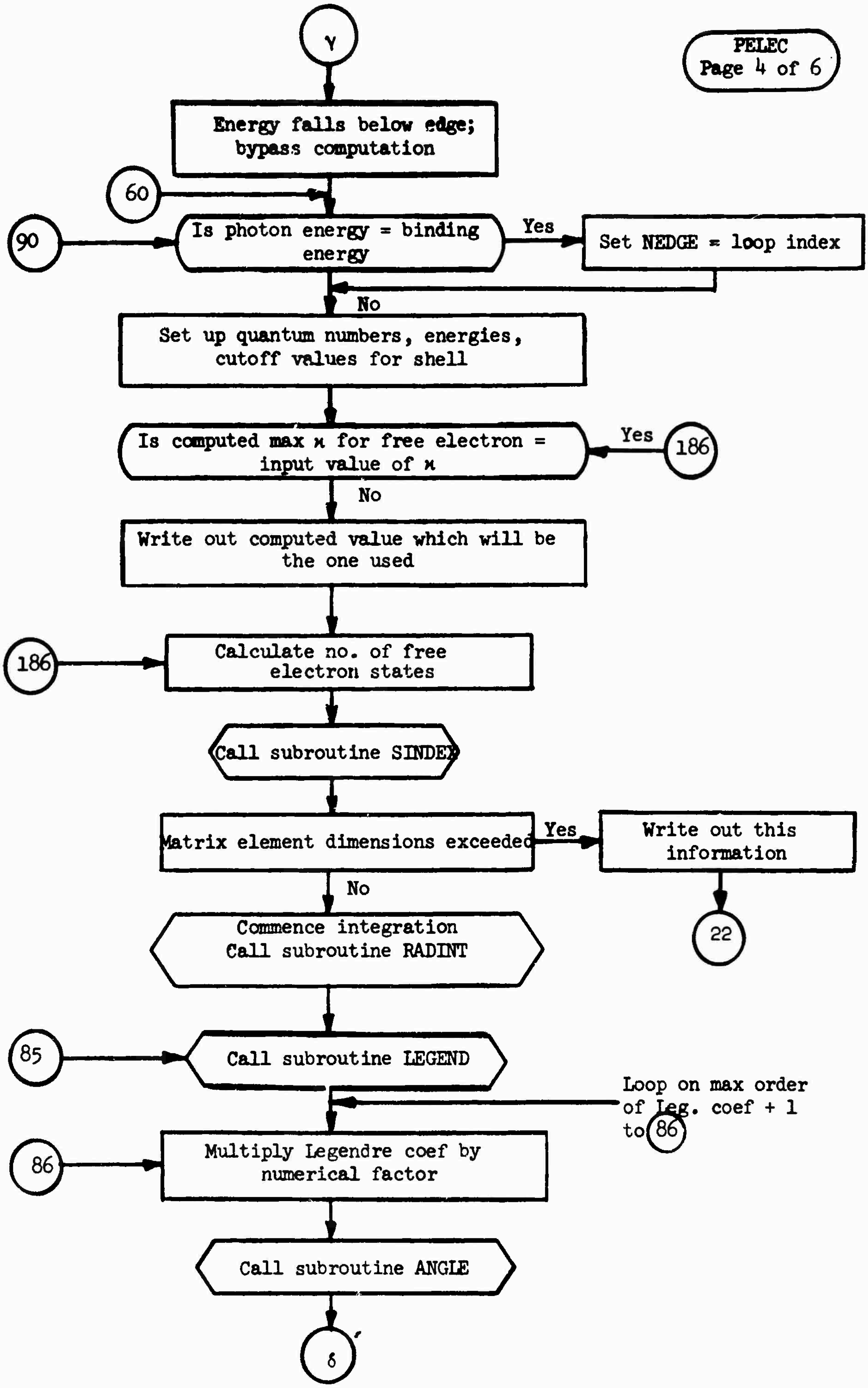
Name	Dimension	Mode	Meaning
ERG	36	R	Binding energy
KJI		I	Number of wavefunction grid points
SECT		R	Total cross section
NTØT		I	Number of matrix elements for shell
SEDGE		R	Cross section jump at edge
LMS		I	Initial max ℓ (in loop reduction)
JMS		I	Initial max order of Legendre coefficient (in loop reduction)
JPS		I	Initial max order of Legendre coefficient + 1 (in loop reduction)
CCM		R	Conversion factor (keV/mc^2)
ALFA		R	Fine structure constant: 1/137.0367
REL		R	Classical electron radius (in cm) x 10^{12}
KB		I	$j_{\kappa'}$
FKB		R	$j_{\kappa'}$

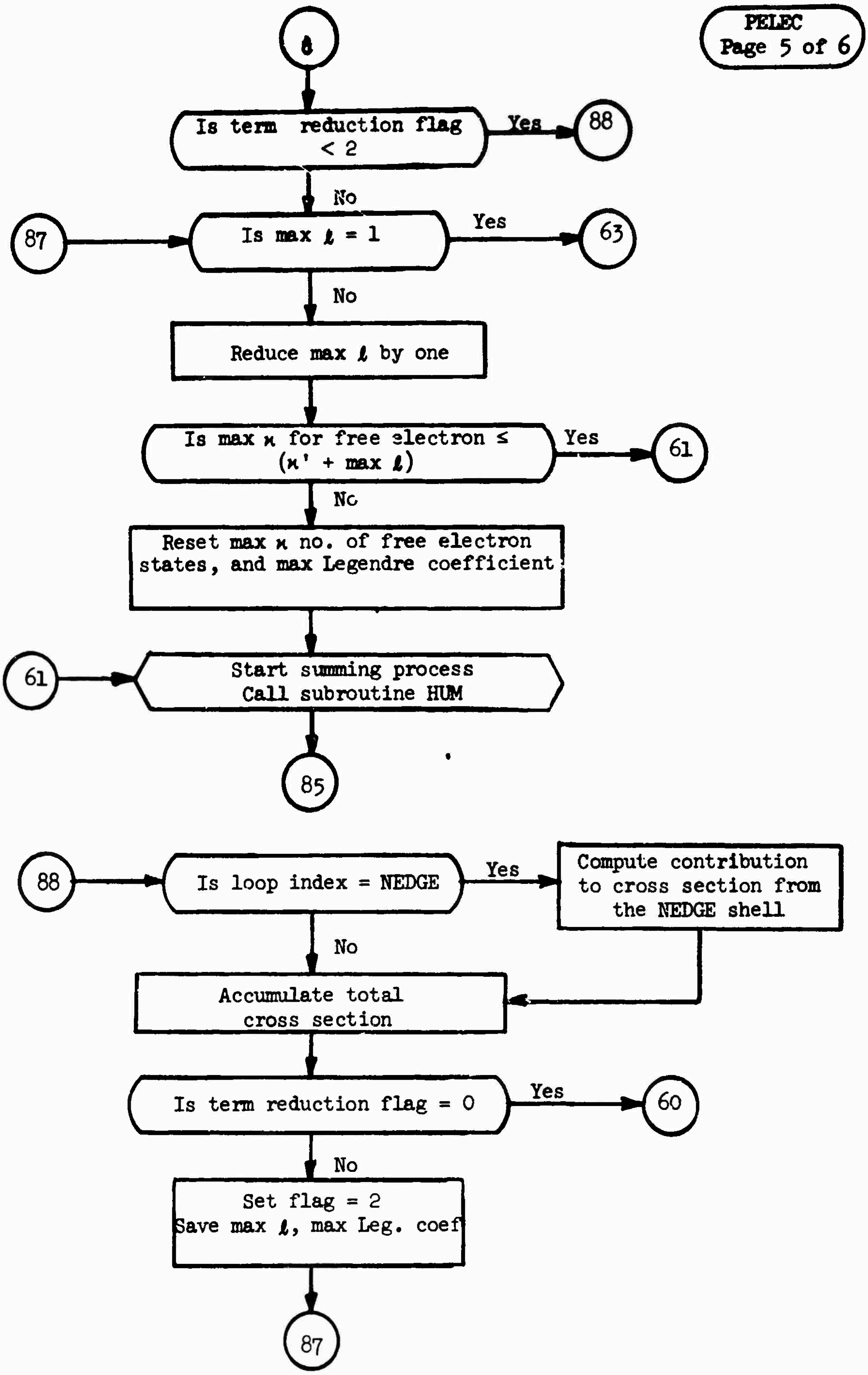
PELEC
Page 1 of 6

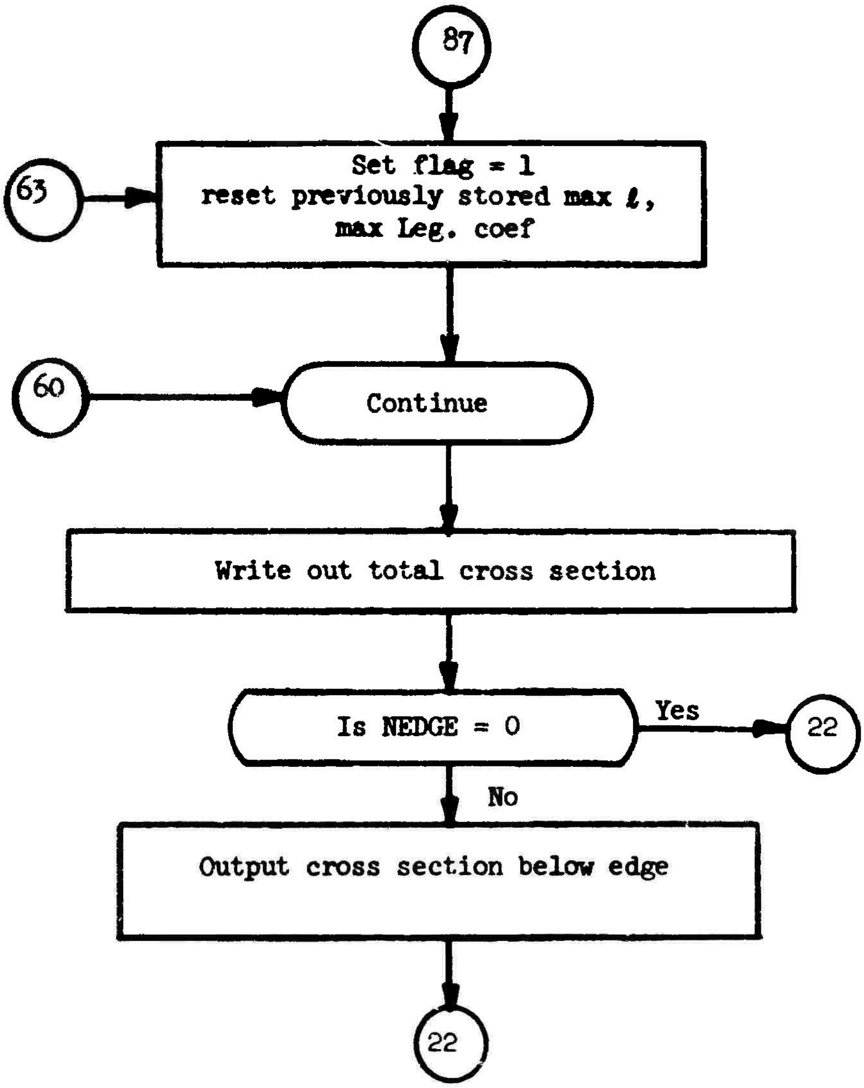

Enter PELEC
Compute & store numerical coefficients for subroutines
22
Read input data
Is NEDGE < 0
Yes
CALL EXIT
No
Write out input data card
Check input data
Are max Leg coef, max ϰ for electron, max ℓ for photon, shell cutoff & shell restart ≥ 0
Yes
29
No
28
Write statement indicating input incorrect
Go to next case
22
29
Is atomic number > 102 or < 2
Yes
36
Write out nuclear charge not found on tape
22
No
Is potential photon energy ≤ 0
Yes
28
No
Is loop for max ℓ reduction ≠ 0 and ≠ 1
Yes
28
No
α


α
Is max j (Legendre coef.) ≤ 24
Yes
111
No
Set max j = 24 and print
111
Is max κ for electron ≤ 12
Yes
112
No
Set max κ = 12 and print
112
Is max ℓ for photon ≤ 12
Yes
116
No
Set max ℓ = 12 and print
116
Write out input quantities with interpretation
35
Read photoelectric tape for element, quantum numbers, binding energies and potential
Does input nuclear charge = nuclear charge read from tape
No
Skip logical records to next element
35
Yes
Write out element located
β


β
Is NEDGE > no. of electron shells
Yes
28
No
Is restart shell number ≥ no. electron shells or cutoff shell number > no. electron shells
Yes
28
No
Read potential
Is NEDGE = 0
Yes
93
No
Reset energy of photon to NEDGE shell binding energy
94
93
Change photon units to mc^2
94
Write out element information from tape
Is cutoff shell = 0
Yes
Set cutoff shell = no. of shells of element
No
Loop on no. of shells to 60
Read wave functions from tape
Is loop index ≤ restart shell no.
Yes
Bypass computation
60
No
Is photon energy ≥ binding energy shell
Yes
90
No
γ


γ
Energy falls below edge; bypass computation
60
90
Is photon energy = binding energy
Yes
Set NEDGE = loop index
No
Set up quantum numbers, energies, cutoff values for shell
Is computed max ϰ for free electron = input value of ϰ
Yes
186
No
Write out computed value which will be the one used
186
Calculate no. of free electron states
Call subroutine SINDEX
Matrix element dimensions exceeded
Yes
Write out this information
22
No
Commence integration Call subroutine RADINT
85
Call subroutine LEGEND
Loop on max order of Leg. coef + 1 to 86
86
Multiply Legendre coef by numerical factor
Call subroutine ANGLE
δ


8
Is term reduction flag < 2
Yes
88
No
87
Is max ℓ = 1
Yes
63
No
Reduce max ℓ by one
Is max $\varkappa$ for free electron ≤ ($\varkappa$' + max ℓ)
Yes
61
No
Reset max $\varkappa$ no. of free electron states, and max Legendre coefficient
61
Start summing process
Call subroutine HUM
85
88
Is loop index = NEDGE
Yes
Compute contribution to cross section from the NEDGE shell
No
Accumulate total cross section
Is term reduction flag = 0
Yes
60
No
Set flag = 2
Save max ℓ, max Leg. coef
87


87
63
Set flag = 1
reset previously stored max ℓ,
max Leg. coef
60
Continue
Write out total cross section
Is NEDGE = 0
Yes
22
No
Output cross section below edge
22

```
$IBFTC PELEC
C     FOR CDC 6600 OPERATION, ADD CARD READING
C     PROGRAM PELEC(INPUT,OUTPUT,TAPE1,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON PI,HALFPI,FOURPI,RAD,SQ2,Q,ZA,ZAZA,EFN,EGN,V,CG(30),GAM(30)PEL00010
      COMMON/BESSEL/FL(15),PC(15),OF(15,15),M1,M2,B(15)                 PEL00020
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),HPEL00030
      COMMON /FAC/FACT(67),RTFAC(95),ROOT(50)                           PEL00040
      COMMON/FIDO/FI(30,15),D(30),JMP,NAME,SHELL,QV,EB,IZ               PEL00050
      COMMON /KUT/RK1(4),RK2(4),RK3(4),RK4(4),K4(4)                     PEL00060
      COMMON /LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB          PEL00070
      COMMON/MAT/SF(30),SG(30),FK(200),FKP(200),SFK(200),SFKP(200),RCUT PEL00080
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)  PEL00090
      COMMON /ONWARD/RX,SCX,GBX,FBX                                     PEL00100
      COMMON /TAPES/X(1500),SCF(1500),FB(1500),GB(1500),GAMB,SCREEN     PEL00110
      COMMON/TRANS/HF(30,15),HFM(30,15),JNG(30),JPS(30)                 PEL00120
      COMMON/VECT/KF(200),KG(200),LBES(200),LBS(200),LKB                PEL00130
      DIMENSION XN(36),XL(36),XJ(36),ERG(36),XZ(36)                     PEL00140
      FL(1) = 1.0                                                       PEL00150
      PC(1) = 1.0                                                       PEL00160
      DO 5 L=2,15                                                       PEL00170
      FL(L) = 2*L-1                                                     PEL00180
      PC(L) = PC(L-1)/FL(L)                                             PEL00190
      DO 5 J=1,15                                                       PEL00200
      FLJ = J*(2*(L+J)-1)                                               PEL00210
    5 OF(L,J) = 1.0/FLJ                                                 PEL00220
      ROOT(1) = 1.0                                                     PEL00230
      DO 10 I=2,50                                                      PEL00240
      FAT = I                                                           PEL00250
   10 ROOT(I) = SQRT (FAT)                                              PEL00260
      FACT(1) = 1.0                                                     PEL00270
      FACT(3) = 1.0                                                     PEL00280
      RTFAC(1) = 1.0                                                    PEL00290
      RTFAC(3) = 1.0                                                    PEL00300
      FAT = 1.0                                                         PEL00310
      DO 15 I=2,33                                                      PEL00320
      FI = I                                                            PEL00330
      FAT = FAT*FI                                                      PEL00340
      FACT(2*I+1) = FAT                                                 PEL00350
   15 RTFAC(2*I+1) = SQRT(FAT)                                          PEL00360
      FAT = 1.0                                                         PEL00370
      DO 20 I=34,47                                                     PEL00380
      FI = I                                                            PEL00390
      FAT = FAT*FI                                                      PEL00400
   20 RTFAC(2*I+1) = SQRT(FAT)*RTFAC(67)                                PEL00410
      SQ2 = 1.0 / ROOT(2)                                               PEL00420
      RK1 (1) = 0.5                                                     PEL00430
      RK1 (2) = 1.0-SQ2                                                 PEL00440
      RK1 (3) = 1.0+SQ2                                                 PEL00450
      RK1 (4) = 1.0/6.0                                                 PEL00460
      RK2 (1) = 2.0                                                     PEL00470
      RK2 (2) = 1.0                                                     PEL00480
      RK2 (3) = 1.0                                                     PEL00490
      RK2 (4) = 2.0                                                     PEL00500
      RK3 (1) = 0.5                                                     PEL00510
      RK3 (2) = 1.0-SQ2                                                 PEL00520
      RK3 (3) = 1.0+SQ2                                                 PEL00530
      RK3 (4) = 0.5                                                     PEL00540
      RK4 (1) = 0.5                                                     PEL00550
      RK4 (2) = 0.0                                                     PEL00560
      RK4 (3) = 0.5                                                     PEL00570
      RK4 (4) = 0.0                                                     PEL00580
      K4(1) = 1                                                         PEL00590
      K4(2) = 0                                                         PEL00600
      K4(3) = 1                                                         PEL00610
      K4(4) = 0                                                         PEL00620
      CCM = 511.0062                                                    PEL00630
      ALFA = 1.0/137.0367                                               PEL00640
      REL = 0.281777                                                    PEL00650
      PI = 3.14159265                                                   PEL00660
      FOURPI = 4.0*PI                                                   PEL00670
      HALFPI = 0.5 * PI                                                 PEL00680
      RAD = PI / 180.0                                                  PEL00690
      FIB = 0.250*PI*REL*REL/ALFA                                       PEL00700
   22 READ (5,25) JM,KMAX,LM,NTAPE,NEDGE,IA,IB,IZ,LOOP,QV,SAVE          PEL00710
   25 FORMAT (9I5,2E15.8)                                               PEL00720
C     ORDINARILY, IA = 0, IB = 0, SAVE = 0.                             PEL00730
C     IB .NE. 0 CUTS OFF CALCULATION AFTER IB SHELLS                    PEL00740
C     IA .GT. 0 RESTARTS AFTER IA SHELLS                                PEL00750
C     SAVE IS THEN CROSS SECTION OF FIRST IA SHELLS READ BACK IN        PEL00760
C     QV ENTERED IN K.E.V., SAVE ENTERED IN BARNS                       PEL00770
C     LOOP = 0 ORDINARILY, LOOP = 1 FOR LM REDUCTION                    PEL00780
C     NEDGE NEGATIVE TERMINATES PROGRAM                                 PEL00790
      IF (NEDGE.LT.0)     CALL EXIT                                     PEL00800
      WRITE (6,27)                                                      PEL00810
   27 FORMAT (1H1///2X,40HINPUT DATA FOR PHOTOELECTRIC CALCULATION //2X,PEL00820
     116HINPUT CARD READS//2X,38HJM  KMAX  LM  NTAPE  NEDGE  IA  IB  IZ,PEL00830
     26H  LOOP,7X,2HQV,10X,4HSAVE //)                                   PEL00840
      WRITE (6,25) JM,KMAX,LM,NTAPE,NEDGE,IA,IB,IZ,LOOP,QV,SAVE         PEL00850
C     DATA CHECKING                                                     PEL00860
      IF (MIN0(JM,KMAX,LM,IA,IB).GE.0)     GO TO 29                     PEL00870
   28 WRITE (6,102)                                                     PEL00880
  102 FORMAT (5X,15HINCORRECT INPUT //)                                 PEL00890
      GO TO 22                                                          PEL00900
   29 IF ((IZ.LT.2).OR.(IZ.GT.102))     GO TO 36                        PEL00910
      IF (QV.LE.0.0)     GO TO 28                                       PEL00920
      IF ((LOOP.NE.0).AND.(LOOP.NE.1))     GO TO 28                     PEL00930
      IF (JM.LE.24)     GO TO 111                                       PEL00940
      JM=24                                                             PEL00950
      WRITE (6,109) JM                                                  PEL00960
  109 FORMAT (/5X56HIN ORDER NOT TO EXCEED DIMENSION JM HAS BEEN REDUCEDPEL00970
     1 TO I3/)                                                          PEL00980
  111 IF (KMAX.LE.12)     GO TO 112                                     PEL00990
      KMAX=12                                                           PEL01000
      WRITE (6,113) KMAX                                                PEL01010
  113 FORMAT (/5X58HIN ORDER NOT TO EXCEED DIMENSION KMAX HAS BEEN REDUCPEL01020
     1ED TO I3/)                                                        PEL01030
  112 IF (LM.LE.12)     GO TO 116                                       PEL01040
      LM=12                                                             PEL01050
      WRITE (6,114) LM                                                  PEL01060
  114 FORMAT (/5X56HIN ORDER NOT TO EXCEED DIMENSION LM HAS BEEN REDUCEDPEL01070
     1 TO I3/)                                                          PEL01080
  116 Z = IZ                                                            PEL01090
      ZA = Z*ALFA                                                       PEL01100
      ZAZA = ZA*ZA                                                      PEL01110
      WRITE (6,50)IZ,QV,KMAX,LM,JM                                      PEL01120
   50 FORMAT (1H1///5X40HINPUT DATA FOR PHOTOELECTRIC CALCULATION //    PEL01130
     114X, I4,17H = NUCLEAR CHARGE /9X,F9.3,16H = PHOTON ENERGY //      PEL01140
     215XI3,25H = MAX KAPPA FOR ELECTRON /15XI3,19H = MAX L FOR PHOTON /PEL01150
     315XI3,26H = MAX J (LEGENDRE COEFF.) //)                           PEL01160
      REWIND NTAPE                                                      PEL01170
      READ (NTAPE) NGRID,ISIXFI,IONE,RSIXFI,RONE,(X(I),I=1,NGRID)       PEL01180
   35 READ (NTAPE) ZTRY,JX,NAME,SCREEN,(XN(I),XL(I),XJ(I),XZ(I),ERG(I), PEL01190
     1I=1,JX)                                                           PEL01200
      IZI = ZTRY + 0.01                                                 PEL01210
      IF ( IZ .EQ. IZT )                     GO TO 40                   PEL01220
      READ (NTAPE) (SCF(I),I=1,NGRID)                                   PEL01230
C     SKIPS  LOGICAL RECORD CONTAINING POTENTIAL                        PEL01240
      DO 99 ISKIP=1,JX                                                  PEL01250
   99 READ (NTAPE) RCUT,KJI,( GB(I),I=1,KJI ), ( FB(I), I=1,KJI )       PEL01260
C     SKIPS UNWANTED  LOGICAL RECORDS TO NEXT ELEMENT                   PEL01270
      IF ( IZT .LE. 102 )                    GO TO 35                   PEL01280
   36 WRITE (6,30) IZ                                                   PEL01290
   30 FORMAT (//30X19HNUCLEAR CHARGE Z = I3,1X18HNOT FOUND ON TAPE.)    PEL01300
      GO TO 22                                                          PEL01310
   40 WRITE (6,45)                                                      PEL01320
   45 FORMAT (/////20X,25HTAPE POSITIONED PROPERLY./////)               PEL01330
      IF (NEDGE.GT.JX)      GO TO 28                                    PEL01340
      IF ((IA.GE.JX).OR.(IB.GT.JX))      GO TO 28                       PEL01350
      READ (NTAPE) (SCF(I),I=1,NGRID)                                   PEL01360
      IF ( NEDGE .EQ. 0 )     GO TO 93                                  PEL01370
      Q = ERG(NEDGE)                                                    PEL01380
      QV = Q * CCM                                                      PEL01390
      GO TO 94                                                          PEL01400
   93 Q = QV / CCM                                                      PEL01410
   94 WRITE (6,173) NAME,IZ,JX,SCREEN,IONE,RONE,ISIXFI,RSIXFI,NGRID,    PEL01420
     1X(NGRID)                                                          PEL01430
  173 FORMAT (//25XA6,1X9H= ELEMENT/28XI3,1X15H= ATOMIC NUMBER/         PEL01440
     129XI2,1X27H= NUMBER OF ELECTRON SHELLS/                           PEL01450
     216XE15.8,1X46H= SCREENING FACTOR OF OUTERMOST BOUND ELECTRON/     PEL01460
     328XI3,1X24H= RADIAL GRID UP TO X = F4.1/                          PEL01470
     427XI4,1X24H= RADIAL GRID UP TO X = F5.1/                          PEL01480
     527XI4,1X19H= TOTAL RADIAL GRID /16XE15.8,1X24H= OUTERMOST RADIAL VPEL01490
     5ALUE //)                                                          PEL01500
      SECT = SAVE                                                       PEL01510
      IF ( IB .EQ. 0 )         IB = JX                                  PEL01520
      DO 60 I = 1, IB                                                   PEL01530
      READ (NTAPE) RCUT,KJI, (GB(L),L=1,KJI),(FB(L),L=1,KJI)            PEL01540
      IF (I.LE.IA)      GO TO 60                                        PEL01550
      IF (Q.GE.ERG(I))      GO TO 90                                    PEL01560
      WRITE (6,95) XN(I)                                                PEL01570
   95 FORMAT (/5X,18HENERGY FALLS BELOW,1XA6,1X4HEDGE//)                PEL01580
      GO TO 60                                                          PEL01590
   90 IF ( Q .EQ. ERG(I) )    NEDGE = I                                 PEL01600
      WRITE (6,80) XN(I),RCUT,KJI                                       PEL01610
   80 FORMAT(1H1/25XA6,1X7H= SHELL/16XE15.8,1X21H= INTEGRATION CUT-OFF/ PEL01620
     127XI4,1X37H= NUMBER OF WAVE FUNCTION GRID POINTS /)               PEL01630
      FST = FIB*XZ(I)/Q                                                 PEL01640
      LKB = XL(I) + 0.01                                                PEL01650
      JKB = 2.0*XJ(I) + 0.01                                            PEL01660
      LMKB = JKB-LKB                                                    PEL01670
      FKB = XJ(I)+0.5                                                   PEL01680
      GAMB = SQRT(FKB**2-ZA*ZA)                                         PEL01690
      KB = FKB + 0.01                                                   PEL01700
      KM = MAX0 ( KB+1, KMAX )                                          PEL01710
      KM = MIN0 ( KM, KB+LM )                                           PEL01720
      IF (KM.EQ.KMAX)      GO TO 186                                    PEL01730
      WRITE (6,187) KM                                                  PEL01740
  187 FORMAT (28XI3,27H = MAX KAPPA FOR THIS SHELL )                    PEL01750
  186 K2M = 2 * KM                                                      PEL01760
      JMP = MIN0 ( JM+1,K2M )                                           PEL01770
      EFN=Q-ERG(I)                                                      PEL01780
      EB=ERG(I)*CCM                                                     PEL01790
      SHELL=XN(I)                                                       PEL01800
      CALL SINDEX                                                       PEL01810
      IF ((NK.LE.200).AND.(NKP.LE.200))      GO TO 83                   PEL01820
      WRITE (6,100) NK,NKP                                              PEL01830
```

```
 100 FORMAT (5X34HMATRIX ELEMENT DIMENSIONS EXCEEDED,14X5HNK = I4,       PEL01840
    16HNKP = I4//30X12HCASE DROPPED //)                                  PEL01850
     GO TO 22                                                            PEL01860
  83 NTOT=NK+NKP                                                         PEL01870
     WRITE (6,1101) NTOT                                                 PEL01880
1101 FORMAT (27X,I4,43H = NUMBER OF MATRIX ELEMENTS FOR THIS SHELL //)   PEL01890
     CALL RADINT                                                         PEL01900
  85 CALL LEGEND                                                         PEL01910
     DO 86 J=1,JMP                                                       PEL01920
  86 D(J)=FST*D(J)                                                       PEL01930
     WRITE (6,2 ) JM, KM, LM                                             PEL01940
   2 FORMAT(1H1///15X,I3,26H = MAX J (LEGENORE COEFF.)/ 15X,I3,          PEL01950
    1      25H = MAX KAPPA FOR ELECTRON/15XI3,19H = MAX L FOR PHOTON/)   PEL01960
     CALL ANGLE                                                          PEL01970
     IF (LOOP.LT.2)      GO TO 88                                        PEL01980
     WRITE (6,62) LM                                                     PEL01990
  62 FORMAT (////50X23H LOOP REDUCTION TO LM = I2)                       PEL02000
  87 IF (LM.EQ.1)      GO TO 63                                          PEL02010
     LM = LM-1                                                           PEL02020
     IF (KM.LE.(KB+LM))      GO TO 61                                    PEL02030
     KM = KB+LM                                                          PEL02040
     K2M = 2*KM                                                          PEL02050
     JMP = MIN0 ( JMP, K2M )                                             PEL02060
     JM = JMP - 1                                                        PEL02070
  61 CALL HUM                                                            PEL02080
     GO TO 85                                                            PEL02090
  88 IF (I.EQ.NEDGE)      SEDGE = FOURPI*D(1)                            PEL02100
     SECT = SECT+FOURPI*D(1)                                             PEL02110
     IF (LOOP.EQ.0)      GO TO 60                                        PEL02120
     LOOP = 2                                                            PEL02130
     LMS = LM                                                            PEL02140
     JMS = JM                                                            PEL02150
     JPS = JMP                                                           PEL02160
     GO TO 87                                                            PEL02170
  63 LOOP = 1                                                            PEL02180
     LM = LMS                                                            PEL02190
     JM = JMS                                                            PEL02200
     JMP = JPS                                                           PEL02210
  60 CONTINUE                                                            PEL02220
     WRITE(6,200) NAME, IZ, QV, SECT                                     PEL02230
 200 FORMAT(1H1////30X,7HELEMENT,27X,13HATOMIC NUMBER//32X,A6,31X,I3///PEL02240
    1 30X,17HPHOTON ENERGY =  ,F12.7,5H  KEV //// 40X,                   PEL02250
    2 22HTOTAL CROSS SECTION =  ,E15.8,7H  BARNS //// )                  PEL02260
     IF (NEDGE.EQ.0)      GO TO 22                                       PEL02270
C    IF AT EDGE, CROSS SECTION BELOW EDGE                                PEL02280
     SECT = SECT-SEDGE                                                   PEL02290
     WRITE (6,75) SECT                                                   PEL02300
  75 FORMAT (/40X27HCROSS SECTION BELOW EDGE = E15.8,6H BARNS/)          PEL02310
     GO TO 22                                                            PEL02320
     END                                                                 PEL02330
```

This page is intentionally left blank.

SUBROUTINE ANGLE

Purpose: This routine computes the angular distribution from the Legendre coefficients of the cross section and a calculation of the Legendre polynomials, and outputs the differential cross-section at two degree intervals.

Method: Legendre polynomials are computed at two degree intervals, then multiplied by the Legendre coefficients of the cross section and finally summed.

Subroutine called: None

Subroutine called by: PELEC

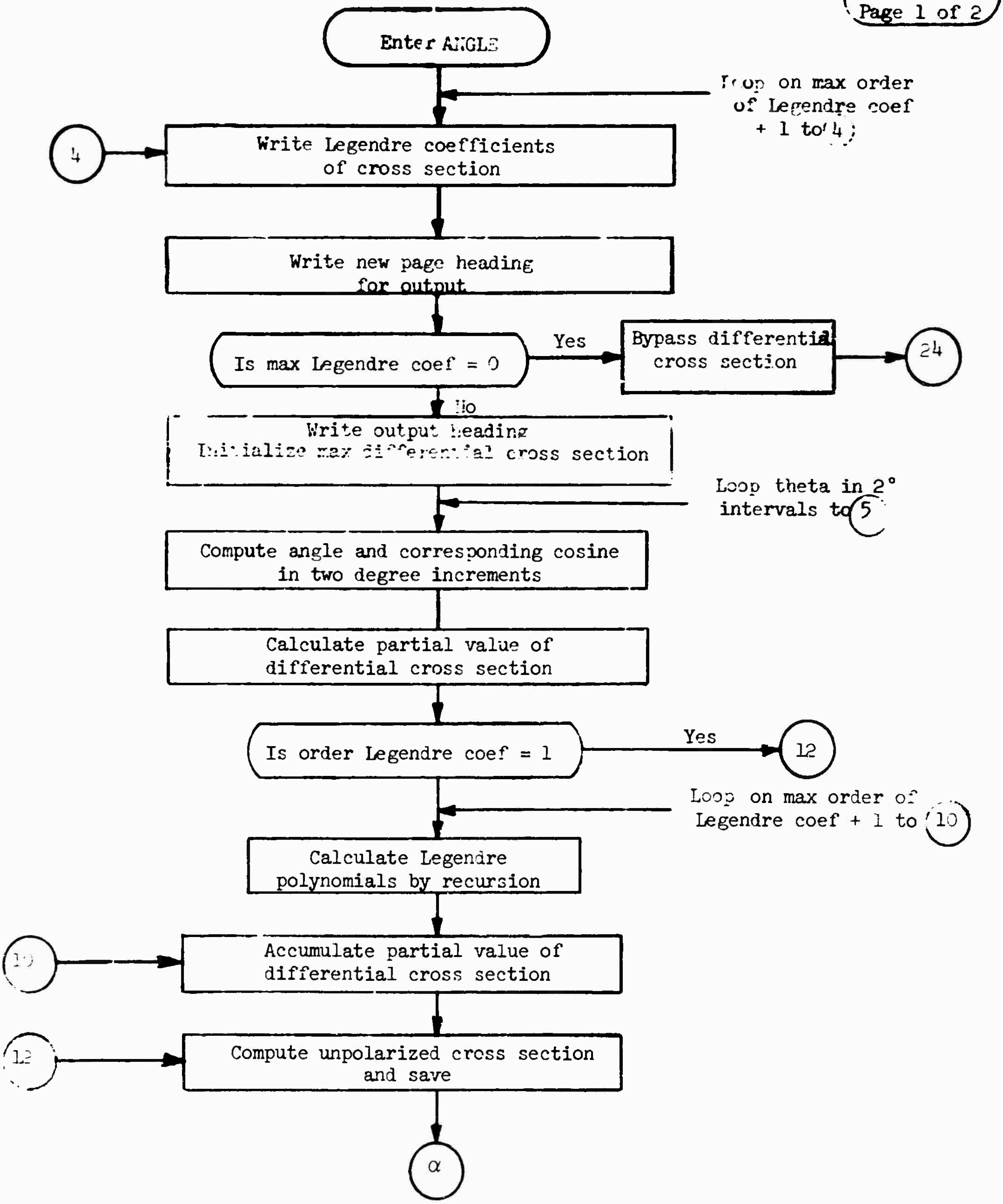
Variables in unlabelled Common: PI, HALFPI, FØURPI, RAD, SQ2, Q, ZA, ZAZA, EFN, V, CG, GAM

Labelled Common: FIDØ

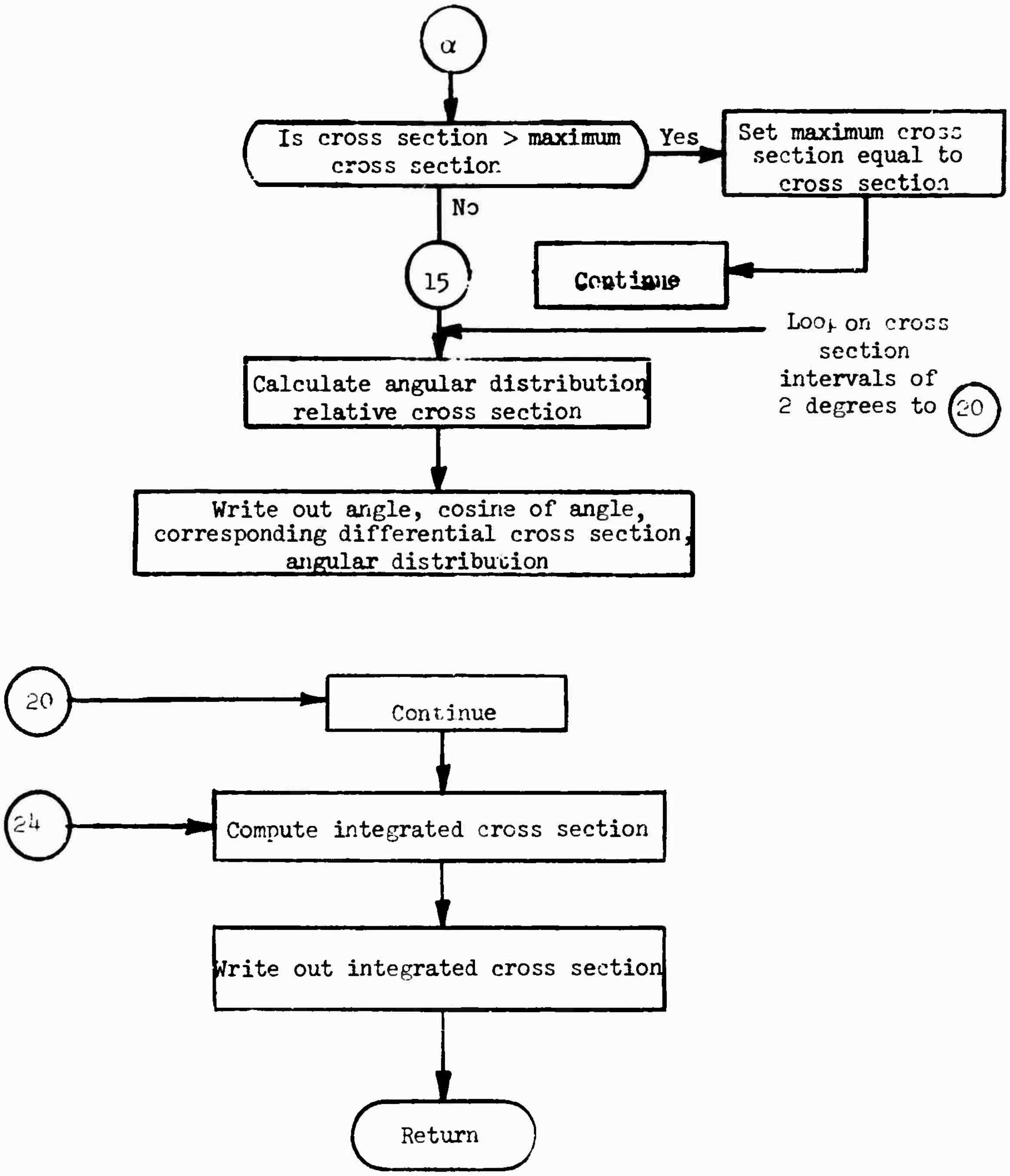
Local Variables:

Name	Dimension	Mode	Meaning
JL		I	Order of Legendre coefficient
EL		R	Electron kinetic energy in keV
SMAX		R	Maximum differential cross section
KMUQ		I	Indexing variable
ITHETA		I	Angle (in degrees) between photon and electron
IT	91	I	Stores angle values
FMUQ		R	Cosine of angle
SIP		R	Unpolarized cross section (in barns/steradian)
PPL		R	Legendre polynomial
SP	91	R	Stored values of unpolarized cross section
AD	91	R	Angular distribution
XSEC		R	Integrated cross section (in barns)
QMU	91	R	Stored values of cosine of angle

ANGLE
Page 1 of 2



Enter ANGLE
Loop on max order of Legendre coef + 1 to 4
4
Write Legendre coefficients of cross section
Write new page heading for output
Is max Legendre coef = 0
Yes
Bypass differential cross section
24
No
Write output heading
Initialize max differential cross section
Loop theta in 2° intervals to 5
Compute angle and corresponding cosine in two degree increments
Calculate partial value of differential cross section
Is order Legendre coef = 1
Yes
12
Loop on max order of Legendre coef + 1 to 10
Calculate Legendre polynomials by recursion
10
Accumulate partial value of differential cross section
12
Compute unpolarized cross section and save
α


α
Is cross section > maximum cross section
Yes
Set maximum cross section equal to cross section
No
15
Continue
Loop on cross section intervals of 2 degrees to 20
Calculate angular distribution relative cross section
Write out angle, cosine of angle, corresponding differential cross section, angular distribution
20
Continue
24
Compute integrated cross section
Write out integrated cross section
Return

```
$IBFTC ANGL
      SUBROUTINE ANGLE
      COMMON PI,HALFPI,FOURPI,RAD,SQ2,Q,ZA,ZAZA,EFN,EGN,V,CG(30),GAM(30)
      COMMON/FIDO/FI(30,15),D(30),JMP,NAME,SHELL,QV,EB,IZ
      DIMENSION IT(91), QMU(91), SP(91), AD(91)
      WRITE (6,2 )
    2 FORMAT(///10X,38HLEGENDRE COEFFICIENTS OF CROSS SECTION//
     1         6X,1HJ,10X,4HD(J) // )
      DO 4 J = 1, JMP
      JL = J - 1
      WRITE (6,3 ) JL, D(J)
    3 FORMAT ( I7,5X, E15.8 )
    4 CONTINUE
      EL = QV - EB
      WRITE (6,5) NAME, IZ, SHELL, EB, QV, EL
    5 FORMAT (1H1////,33X,7HELEMENT,7X,13HATOMIC NUMBER, 13X,5HSHELL//
     1         35X,A6,11X,I3,17X,A6///16X,14HBINDING ENERGY,17X,
     2         13HPHOTON ENERGY,11X,23HELECTRON KINETIC ENERGY//
     3         3(14X,F12.7,4H KEV) /// )
      IF ( JMP .EQ. 1 )           GO TO 24
      WRITE (6,8)
    8 FORMAT(25X,43HUNPOLARIZED CROSS SECTION (BARNS/STERADIAN) //
     1         16X,5HTHETA,15X,9HCOS THETA,18X,13HCROSS SECTION, 16X,
     2         10HANG. DIST. // )
      SMAX = 0.0
      DO 15 KMUQ = 1, 91
      ITHETA = 2 * ( KMUQ - 1 )
      IT(KMUQ) = ITHETA
      THE=RAD*FLOAT (ITHETA)
      FMUQ=COS (THE)
      QMU(KMUQ) = FMUQ
      PMI=1.0
      PN=FMUQ
      SIP = D(1) + FMUQ * D(2)
      IF (JMP.EQ.2)      GO TO 12
      DO 10 J=3,JMP
      FN=J-2
      PPL = ( PN*FMUQ*(2.0*FN+1.0) - PMI*FN ) / ( FN+1.0 )
      PMI=PN
      PN=PPL
   10 SIP = SIP + PPL * D(J)
   12 SP(KMUQ) = SIP
      IF ( SP(KMUQ) .GT. SMAX )           SMAX = SP(KMUQ)
   15 CONTINUE
      DO 20 KMUQ = 1, 91
      AD(KMUQ) = SP(KMUQ) / SMAX
      WRITE ( 6,18) IT(KMUQ), QMU(KMUQ), SP(KMUQ), AD(KMUQ)
   18 FORMAT ( 15X,I5,15X,F10.7,17X ,E15.8 , 15X, F10.7 )
   20 CONTINUE
   24 XSEC = FOURPI * D(1)
      WRITE (6,25) XSEC
   25 FORMAT (/////30X27HINTEGRATED CROSS SECTION = E15.8, 6H BARNS //)
      RETURN
      END
```

SUBROUTINE CØEFS

Purpose: Computes Clebsch-Gordan coefficients.

Method: For computer computation economy, the requisite square roots of integers, factorials and square roots of factorials are stored in common. The routine uses explicit algebraic expressions for Clebsch-Gordan coefficients whose smallest angular momentum value is two or less (with appropriate p tation of indices). Otherwise the general formula is used, with a specialization for the parity Clebsch-Gordan coefficients. The input variables are double the angular momenta quantities, in order to use them in integer mode.

Subroutine called: None

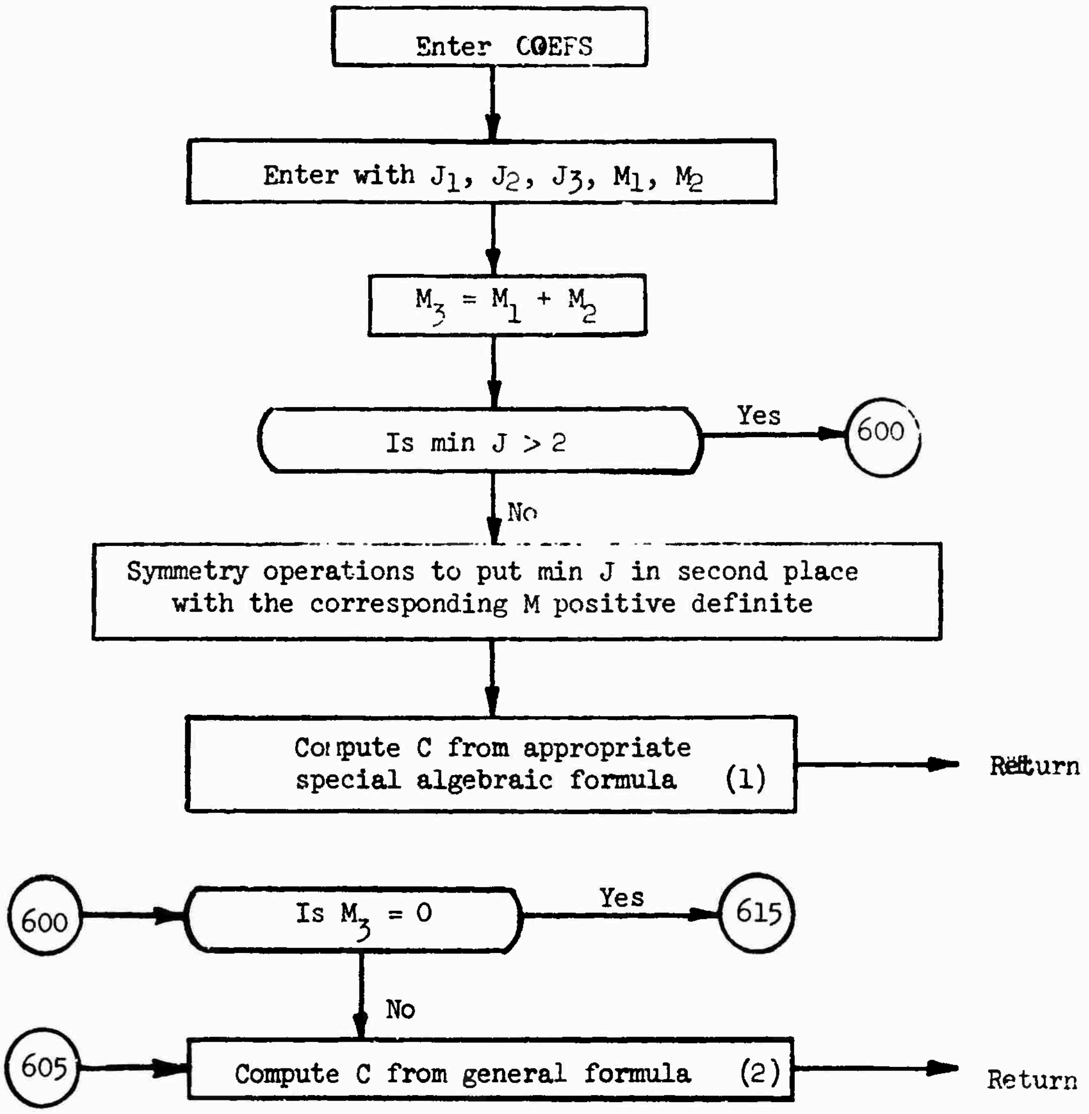
Subroutine called by: FILL, HUM, MUSS

Labelled common: FAC

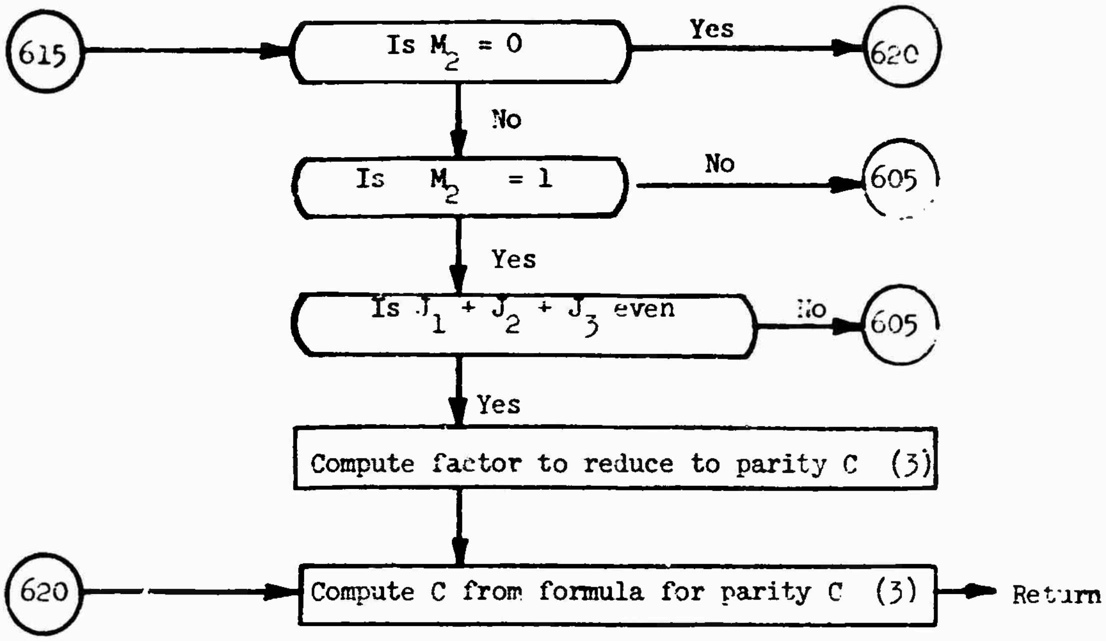
Argument sequence: (J1, J2, J3, M1, M2, C)

Argument List:

Name	Dimension	Mode	Meaning
J1, J2, J3		I	Angular momenta
M1, M2		I	Magnetic quantum numbers
C		R	Clebsch-Gordan coefficient


Enter COEFS
Enter with J_1, J_2, J_3, M_1, M_2
$M_3 = M_1 + M_2$
Is min J > 2
Yes
600
No
Symmetry operations to put min J in second place with the corresponding M positive definite
Compute C from appropriate special algebraic formula (1)
Return
600
Is $M_3 = 0$
Yes
615
No
605
Compute C from general formula (2)
Return

COEFS
Page 2 of 2


615
Is M_2 = 0
Yes
620
No
Is M_2 = 1
No
605
Yes
Is $J_1 + J_2 + J_3$ even
No
605
Yes
Compute factor to reduce to parity C (3)
620
Compute C from formula for parity C (3)
Return

References:

(1) E. V. Condon and G. H. Shortley, Theory of Atomic Spectra, (Cambridge University Press, 1935).

(2) G. Racah, Phys. Rev. 62, 438 (1942).

(3) L. C. Biedenharn and M. E. Rose, Rev. Mod. Phys. 25, 729 (1953).

```
$IBFTC COEF
      SUBROUTINE COEFS ( J1,J2,J3,M1,M2,C )                             COF00010
      COMMON /FAC/F(67),RT(95),R(50)                                    COF00020
      M3  =  M1 + M2                                                    COF00030
      C = 0.0                                                           COF00040
      SIGM = 1.0                                                        COF00050
      JMIN  =   MIN0 (J1,J2,J3)                                         COF00060
      IF ( JMIN .GT. 4  )        GO TO 600                              COF00070
      IF ( JMIN .EQ. J2 )       GO TO 220                               COF00080
      IF ( JMIN .EQ. J3 )        GO TO 230                              COF00090
  210 L1 = J2                                                           COF00100
      L2  = J1                                                          COF00110
      L3  = J3                                                          COF00120
      LM1 = -M2                                                         COF00130
      LM2 = -M1                                                         COF00140
      LM3 = -M3                                                         COF00150
      GO  TO       240                                                  COF00160
  220 L1 = J1                                                           COF00170
      L2  = J2                                                          COF00180
      L3  = J3                                                          COF00190
      LM1 = M1                                                          COF00200
      LM2 = M2                                                          COF00210
      LM3 = M3                                                          COF00220
      GO  TO      240                                                   COF00230
  230 L1  = J1                                                          COF00240
      L2 = J3                                                           COF00250
      L3  = J2                                                          COF00260
      LM1 = M1                                                          COF00270
      LM2 = -M3                                                         COF00280
      LM3 = -M2                                                         COF00290
      SIGM = R(L2+1)/R(L3+1)                                            COF00300
      IF ( MOD ( J1 - M1 , 4 ) .NE. 0 )                SIGM = - SIGM    COF00310
  240 IF ( LM2 ) 245, 250, 250                                          COF00320
  245 LM1 = - LM1                                                       COF00330
      LM2 = - LM2                                                       COF00340
      LM3 = - LM3                                                       COF00350
      IF ( MOD ( L1 + L2 - L3 , 4 ) .NE. 0 )           SIGM = - SIGM    COF00360
  250 JMIN = JMIN+1                                                     COF00370
      K = L1+LM3                                                        COF00380
      L = L1-LM3                                                        COF00390
      GO  TO          (255,260,300,700,400),    JMIN                    COF00400
  255 IF (L1-L3) 800,256,800                                            COF00410
  256 IF (LM1-LM3) 800,257,800                                          COF00420
  257 C  =  SIGM                                                        COF00430
      GO TO 800                                                         COF00440
  260 IF (L3-L1-LM2) 265,280,270                                        COF00450
  265 SIGM = -SIGM                                                      COF00460
  270 K = L                                                             COF00470
  280 KP1 = K + 1                                                       COF00480
      C= R(KP1)/R(2*L1+2)                                               COF00490
  290 C = SIGM*C                                                        COF00500
      GO  TO        800                                                 COF00510
  300 IF (L3-L1) 305,310,315                                            COF00520
  305 IF (LM2) 800,325,330                                              COF00530
  310 IF (LM2) 800,340,345                                              COF00540
  315 IF (LM2) 800,355,360                                              COF00550
  325 C = -R(L)*R(K)/(R(2*L1)*R(L1+1))                                  COF00560
      GO  TO       380                                                  COF00570
  330 C = R(L)*R(L+2)/(2.0*R(L1)*R(L1+1))                               COF00580
      GO  TO           380                                              COF00590
  340 C = FLOAT(LM3)/(R(L1)*R(L1+2))                                    COF00600
      GO  TO          380                                               COF00610
```

```
345 LOX = L+2                                                           COF00620
    C = -R(K)*R(LOX)/(R(2*L1)*R(L1+2))                                  COF00630
    GO  TO      380                                                     COF00640
355 LOX = L+2                                                           COF00650
    LAX = K+2                                                           COF00660
    C = R(LOX)*R(LAX)/(R(2*L1+2)*R(L1+2))                               COF00670
    GO  TO        380                                                   COF00680
360 C = R(K)*R(K+2)/(2.0*R(L1+1)*R(L1+2))                               COF00690
380 C = SIGM*C                                                          COF00700
    GO  TO       800                                                    COF00710
400 M = LM2/2+1                                                         COF00720
    J = (L3-L1)/2 +3                                                    COF00730
    GO  TO                  (480,510,540),         M                    COF00740
480 GO  TO         (485,490,495,500,505),          J                    COF00750
485 C = R(3)*R(L)*R(L-2)*R(K)*R(K-2)/(R(8)*R(L1-2)*R(L1-1)*R(L1)        COF00760
   1*R(L1+1))                                                           COF00770
    GO  TO       575                                                    COF00780
490 C = -0.5*FLOAT(LM3)*R(6)*R(L)*R(K)/(R(L1)*R(L1-2)*R(L1+1)*R(L1+2))COF00790
    GO  TO            575                                               COF00800
495 C = 0.5 *FLOAT(3*LM3*LM3-L1*(L1+2))                                 COF00810
   1                                       /(R(L1)*R(L1-1)*R(L1+2)*R(L1+3))COF00820
    GO  TO        575                                                   COF00830
500 LOX = L+2                                                           COF00840
    LAX = K+2                                                           COF00850
    C = 0.5*FLOAT(LM3)                                                  COF00860
   1                 *R(6)*R(LOX)*R(LAX)/(R(L1)*R(L1+1)*R(L1+2)*R(L1+4)) COF00870
    GO  TO           575                                                COF00880
505 LOX = L+4                                                           COF00890
    LAX = L+2                                                           COF00900
    LLX =K+4                                                            COF00910
    LXX = K+2                                                           COF00920
    C = R(3)*R(LOX)*R(LAX)*R(LLX)*R(LXX)/(R(8)*R(L1+1)*R(L1+2)*R(L1+3)COF00930
   1*R(L1+4))                                                           COF00940
    GO  TO     575                                                      COF00950
510 GO  TO          (515,520,525,530,535),         J                    COF00960
515 C = -R(L+2)*R(L)*R(L-2)*R(K-2)/(2.0*R(L1-2)*R(L1-1)*R(L1)*R(L1+1))COF00970
    GO  TO       575                                                    COF00980
520 C = 0.5*FLOAT(L1+2*LM3-2)                                           COF00990
   1                        *R(L+2)*R(L)/(R(L1)*R(L1-2)*R(L1+1)*R(L1+2)) COF01000
    GO  TO           575                                                COF01010
525 LOX = L+2                                                           COF01020
    C = (1.0-FLOAT(LM3))*R(3)*R(LOX)*R(K)/(R(2*L1)*R(L1-1)*R(L1+2)      COF01030
   1*R(L1+3))                                                           COF01040
    GO  TO              575                                             COF01050
530 C=.5*FLOAT(2*LM3-L1-4)                                              COF01060
   1                        *R(K+2)*R(K)/(R(L1)*R(L1+1)*R(L1+2)*R(L1+4)) COF01070
    GO  TO       575                                                    COF01080
535 LOX = L+4                                                           COF01090
    C=R(LOX)*R(K+4)*R(K+2)*R(K)/(2.0*R(L1+1)*R(L1+2)*R(L1+3)*R(L1+4)) COF01100
    GO  TO       575                                                    COF01110
540 GO  TO          (545,550,555,560,565) ,        J                    COF01120
545 C = R(L-2)*R(L)*R(L+2)*R(L+4)/(4.0*R(L1)*R(L1-2)*R(L1-1)*R(L1+1)) COF01130
    GO  TO       575                                                    COF01140
550 C= -R(K-2)*R(L)*R(L+2)*R(L+4)/(2.0*R(L1)*R(L1-2)*R(L1+2)*R(L1+1)) COF01150
    GO  TO        575                                                   COF01160
555 LOX = L+2                                                           COF01170
    LAX = L+4                                                           COF01180
    C = R(3)*R(K-2)*R(K)*R(LOX)*R(LAX)/(R(8)*R(L1)*R(L1-1)*R(L1+2)      COF01190
   1*R(L1+3))                                                           COF01200
    GO  TO        575                                                   COF01210
560 LOX = L+4                                                           COF01220
    C = -R(K-2)*R(K)*R(K+2)*R(LOX)/(2.0*R(L1)*R(L1+2)*R(L1+4)*R(L1+1))COF01230
```

```
      GO  TO      575                                                      COF01240
  565 C=R(K-2)*R(K)*R(K+2)*R(K+4)/(4.0*R(L1+1)*R(L1+2)*R(L1+3)*R(L1+4))   COF01250
  575 C = SIGM*C                                                           COF01260
      GO  TO       800                                                     COF01270
  700 M = (LM2+1)/2                                                        COF01280
      J = (L3-L1+5)/2                                                      COF01290
      GO TO ( 710, 740 ) ,    M                                            COF01300
  710 GO TO ( 720, 725, 730, 735 ) ,    J                                  COF01310
  720 C = R(3)*R(K-1)*R(L-1)*R(L+1)/(R(8)*R(L1)*R(L1-1)*R(L1+1))           COF01320
      GO TO 780                                                            COF01330
  725 C = -FLOAT((L1+3*LM3-1)/2)*R(L+1)/(R(2)*R(L1-1)*R(L1+1)*R(L1+2))     COF01340
      GO TO 780                                                            COF01350
  730 KP1 = K + 1                                                          COF01360
      C = -FLOAT((3-3*LM3+L1)/2)*R(KP1)/(R(2)*R(L1)*R(L1+1)*R(L1+3))       COF01370
      GO TO 780                                                            COF01380
  735 LOX = L+3                                                            COF01390
      KP1 = K + 1                                                          COF01400
      KP3 = K + 3                                                          COF01410
      C = R(3)*R(KP1)*R(KP3)*R(LOX)/(R(8)*R(L1+1)*R(L1+2)*R(L1+3))         COF01420
      GO TO 780                                                            COF01430
  740 GO TO ( 750, 755, 760, 765 ) ,    J                                  COF01440
  750 C = -R(L-1)*R(L+1)*R(L+3)/(R(8)*R(L1)*R(L1-1)*R(L1+1))               COF01450
      GO TO 780                                                            COF01460
  755 LOX = L+3                                                            COF01470
      LAX = L+1                                                            COF01480
      C = R(3)*R(K-1)*R(LAX)*R(LOX)/(R(8)*R(L1-1)*R(L1+1)*R(L1+2))         COF01490
      GO TO 780                                                            COF01500
  760 LOX = L+3                                                            COF01510
      C = -R(3)*R(K-1)*R(K+1)*R(LOX)/(R(8)*R(L1)*R(L1+1)*R(L1+3))          COF01520
      GO TO 780                                                            COF01530
  765 C = R(K-1)*R(K+1)*R(K+3)/(R(8)*R(L1+1)*R(L1+2)*R(L1+3))              COF01540
  780 C = C*SIGM                                                           COF01550
      GO TO 800                                                            COF01560
C     THIS IS THE BEGINNING OF COMPUTATION OF C-COEFFICIENT                COF01570
C     USING THE GENERAL EXPRESSION.                                        COF01580
  600 L1 = J1+J2-J3+1                                                      COF01590
      L2 = J1-J2+J3+1                                                      COF01600
      L3 = -J1+J2+J3+1                                                     COF01610
      L10 = J1+J2+J3+3                                                     COF01620
      IF ( M3 .EQ. 0 )              GO TO 615                              COF01630
  605 L4 = J1+M1+1                                                         COF01640
      L5 = J1-M1+1                                                         COF01650
      L6 = J2+M2+1                                                         COF01660
      L7 = J2-M2+1                                                         COF01670
      L8 = J3+M3+1                                                         COF01680
      L9 = J3-M3+1                                                         COF01690
      ST = RT(L10) / (   RT(L1)*RT(L4)*RT(L5) )                            COF01700
      ST = ST / ( RT(L6) * RT(L7) )                                        COF01710
      ST = ST / (  R(J3+1)*RT(L2)*RT(L3) )                                 COF01720
      ST = ST / (  RT(L8) * RT(L9) )                                       COF01730
      N7 = L1-L7                                                           COF01740
      N4 = L1-L4                                                           COF01750
      MIN = MAX0 (0,N4,N7)                                                 COF01760
      MAX = MIN0 (L1,L5,L6)                                                COF01770
      IF (MOD(MIN,4).NE.0)      SIGM=-1.0                                  COF01780
      MIN = MIN+1                                                          COF01790
      N1 = L1+1                                                            COF01800
      N5 = L5+1                                                            COF01810
      N6 = L6+1                                                            COF01820
      SUM = 0.0                                                            COF01830
      DO 610 LZ=MIN,MAX,2                                                  COF01840
      N1L = N1-LZ                                                          COF01850
      N5L = N5-L2
      N6L = N6-L2
      N4L =-N4+L2
      N7L =-N7+L2
      TERM = ST * F(L2) * F(N1L) * F(N5L) * F(N6L) * F(N4L) * F(N7L)
      C = C + SIGM / TERM
  610 SIGM = -SIGM
      GO TO 800
  615 IF ( M2 .EQ. 0 )             GO TO 620
      IF ( IABS(M2) .NE. 2 )       GO TO 605
      JMOD = MOD ( (L10 + 1 ), 4 )
      IF ( JMOD .NE. 0 )           GO TO 605
      ST = J3 * (J3 + 2) - J1 * (J1 + 2) - J2 * (J2 + 2)
      SIGM = 0.5 * SIGM * ST / ( R(J1)*R(J1 + 2)*R(J2)*R(J2 + 2) )
  620 JMOD = MOD (   ( L1 - 1 ), 8   )
      IF ( JMOD .NE. 0 )           SIGM = - SIGM
      L4 = ( L1  + 1 ) / 2
      L5 = ( L2  + 1 ) / 2
      L6 = ( L3  + 1 ) / 2
      L7 = ( L10 - 1 ) / 2
      C = SIGM * R(J3 + 1) * F(L7) / (  F(L4)*F(L5)*F(L6)   )
      C = C * RT(L1) * RT(L2) * RT(L3) /  RT(L10)
  800 RETURN
      END
```

This page is intentionally left blank.

SUBROUTINE DERIV

Purpose: Computes for the Runge-Kut a integration the derivatives of the bound state wavefunctions and the integrands of the matrix elements up to one-half Bohr radius.

Method: Calculates the derivative of the radial components from the coupled Dirac radial equations. In evaluating the integrand of the matrix elements the r^{γ} factor is restored if $r < 1$.

Subroutine called: SPHBES

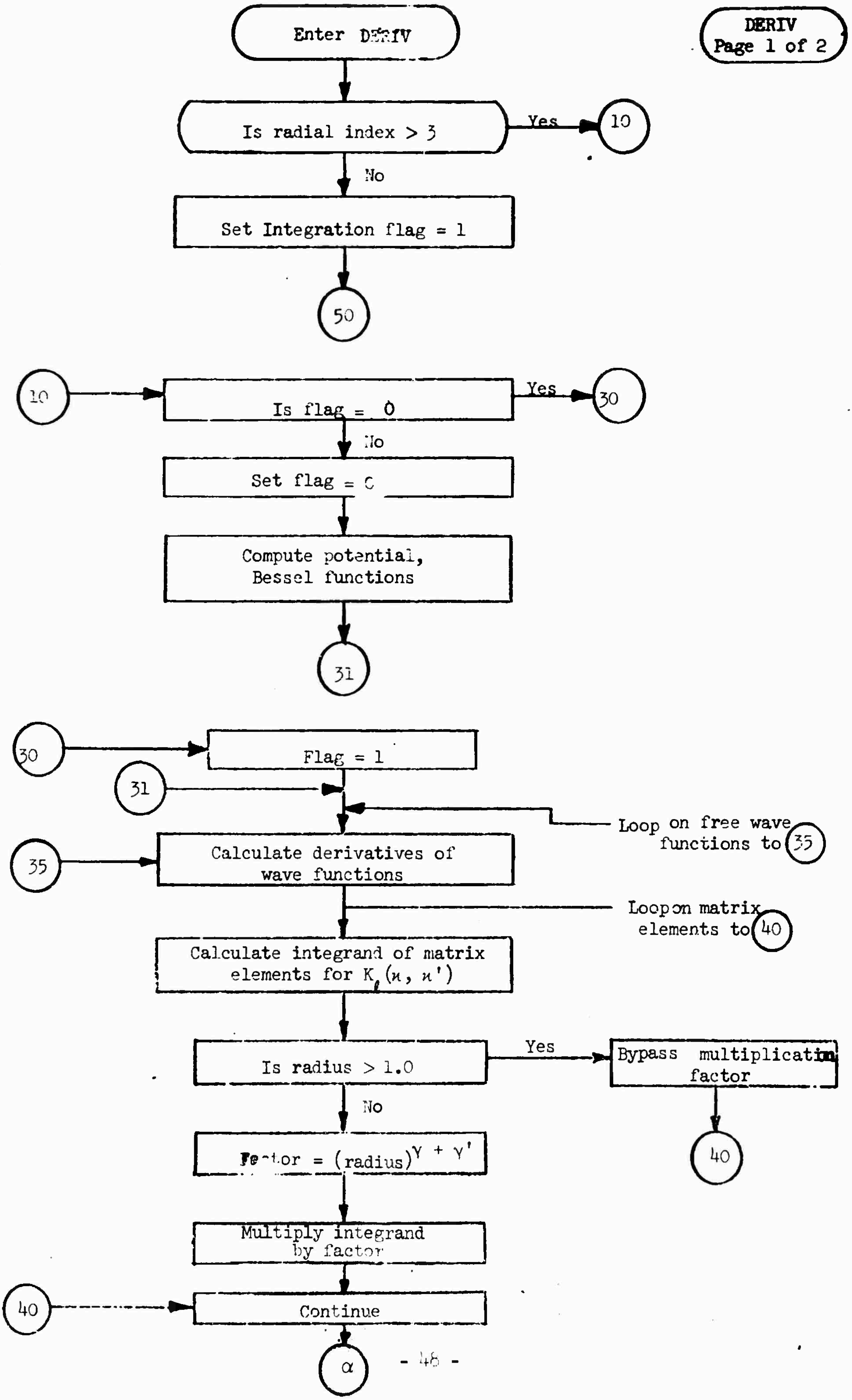
Subroutine called by: RKUT, RADINT

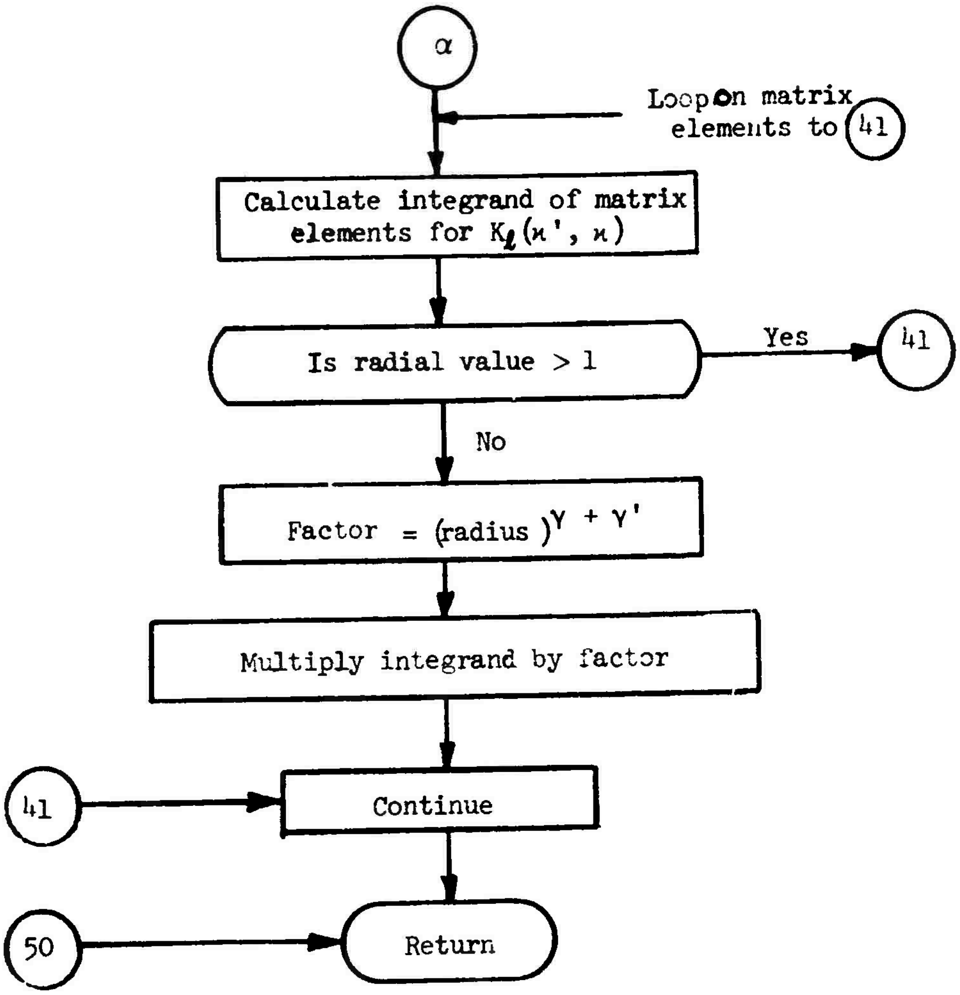
Variables in unlabelled Common: PI, HALFPI, FØURPI, RAD, SQ2, Q, ZA, ZAZA, EFN, EGN, V, CG, GAM

Labelled Common: BESSEL, DEFUNC, LIMIT, TAPES, VECT

Local Variables:

Name	Dimension	Mode	Meaning
Z		R	Photon momentum * radius
N		I	Indexing variable
A		R	Sum of gammas of bound and free state electron


DERIV
Page 1 of 2
Enter DERIV
Is radial index > 3
Yes
10
No
Set Integration flag = 1
50
10
Is flag = 0
Yes
30
No
Set flag = 0
Compute potential,
Bessel functions
31
30
Flag = 1
31
Loop on free wave
functions to 35
35
Calculate derivatives of
wave functions
Loop on matrix
elements to 40
Calculate integrand of matrix
elements for $K_\ell(\kappa, \kappa')$
Is radius > 1.0
Yes
Bypass multiplication
factor
40
No
Factor = $(\text{radius})^{\gamma + \gamma'}$
Multiply integrand
by factor
40
Continue
α


α
Loop on matrix elements to 41
Calculate integrand of matrix elements for $K_\ell(\kappa', \kappa)$
Is radial value > 1
Yes
41
No
Factor = (radius)$^{\gamma + \gamma'}$
Multiply integrand by factor
41
Continue
50
Return

```
$IBFTC DERI
      SUBROUTINE DERIV
C                    COMPUTES DERIVATIVES
      COMMON PI,HALFPI,FOURPI,RAD,SQ2,Q,ZA,ZAZA,EFN,EGN,V,CG(30),GAM(30)
      COMMON/BESSEL/FL(15),PC(15),OF(15,15),M1,M2,B(15)
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),H
      COMMON /LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB
      COMMON /TAPES/X(1500),SCF(1500),FB(1500),GB(1500),GAMB,SCREEN
      COMMON/VECT/KF(200),KG(200),LBES(200),LBS(200),LKB
      IF (NTAB.GT.3)      GO TO 10
      NEW = 1
      GO TO 50
   10 IF (NEW.EQ.0)      GO TO 30
      NEW = 0
      V=-SCF(NTAB)/X(NTAB)
      Z = Q*X(NTAB)
      CALL SPHBES (Z)
      GO TO 31
   30 NEW = 1
   31 DO 35 N=1,K2M
      DF(N) = CF(N)*F(N)/X(NTAB)-(EFN-V)*G(N)
   35 DG(N) = CG(N)*G(N)/X(NTAB)+(EGN-V)*F(N)
      DO 40 N=1,NK
      I = KG(N)
      L = LBES(N)
      DFK(N)=B(L)*G(I)*FB(NTAB)
      IF (X(NTAB).GT.1.0)      GO TO 40
      A = GAM(I)+GAMB
      DFK(N) = DFK(N)*(X(NTAB))**A
   40 CONTINUE
      DO 41 N=1,NKP
      I = KF(N)
      L = LBS(N)
      DFKP(N)=B(L)*F(I)*GB(NTAB)
      IF (X(NTAB).GT.1.0)      GO TO 41
      A = GAM(I)+GAMB
      DFKP(N) = DFKP(N)*X(NTAB)**A
   41 CONTINUE
   50 RETURN
      END
```

SUBROUTINE FILL

Purpose: Computes $\phi(\kappa,\lambda)$

Method: $\phi(\kappa,\lambda)$ is computed in two passes, one for $K_\ell(\kappa\ \kappa')$ contributions the other for $K_\ell(\kappa'\ \kappa)$ and then the two pieces are added in subroutine HUM to yield ϕ. The selection rules are examined for each matrix element contribution to determine what values of $(\lambda - \ell)$ are allowed. If $\lambda = \ell \pm 1$, f can assume only the one value $f = \lambda \mp 1/2$; if $\lambda = \ell$ both the preceding values of f are possible. In the final run-through the remaining selection rules required for R are checked, R is computed from explicit algebraic expressions for the Racah coefficient with $J_5 = 1/2$, the corresponding Clebsch-Gordan coefficient called and $\phi(\kappa,\lambda)$ formed with appropriate ℓ summation. The indices used for ϕ are λ and K (K is a positive integer uniquely related to κ).

Subroutine called: CØEFS

Subroutine called by: HUM

Labelled Common: FAC, LIMIT, QUANT

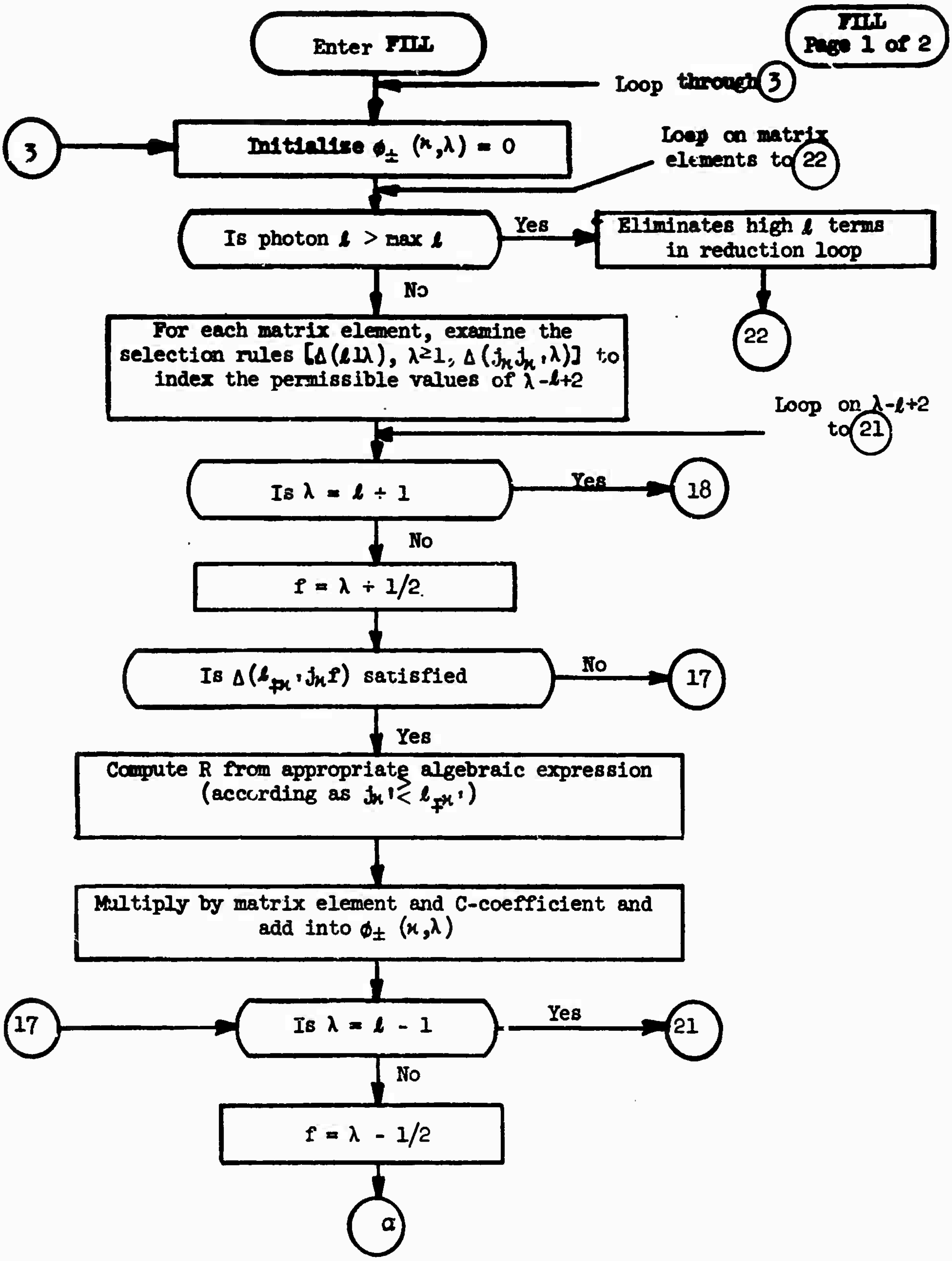
Argument sequence: (TK, KW, LB, NT, LP, FI)

Argument List:

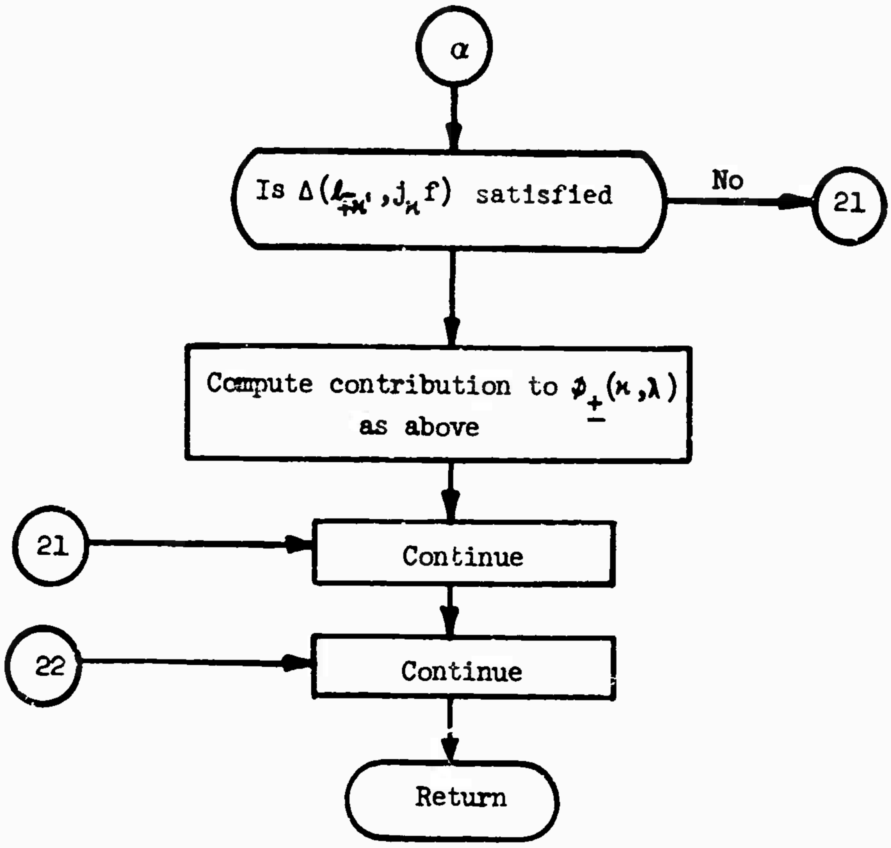
Name	Dimension	Mode	Meaning
TK	200	R	Matrix elements
KW	200	I	Index for κ values in matrix elements
LB	200	I	Photon angular momentum + 1 for matrix elements
NT		I	Number of matrix elements $K_\ell(\kappa\ \kappa')$ or $K_\ell(\kappa'\kappa)$
LP		I	$\ell_{-\kappa'}$ or $\ell_{\kappa'}$
FI	30 x 15	R	$\phi_+(\kappa,\lambda)$ or $\phi_-(\kappa,\lambda)$

Local Variables:

Name	Dimension	Mode	Meaning
LMP		I	Max photon angular momentum + 1
LA		I	λ
K		I	Index of free electron state
LD		I	Twice bound electron $\ell_{\mp\kappa'}$
N		I	Index of matrix element
L		I	Photon angular momentum
JC		I	Twice free electron j_{κ}
NA, NB		I	Loop index for the range of $(\lambda - \ell)$
LAM		I	$2 * \lambda$
LEF		I	$2 * (\lambda \pm 1/2)$
NL		I	$(\lambda - \ell) + 2$
IR, IS		I	Terms in R
R		R	$R\ (j_{\kappa'}, \ell_{\mp\kappa'}, \lambda f j_{\kappa})$
C		R	Clebsch-Gordan coefficient
JPL		I	$2 \times j_{\kappa'} + 2 \times j_{\kappa}$ - 2 x photon angular momentum
JML		I	$\lvert 2 \times j_{\kappa'} - 2 \times j_{\kappa} \rvert$ - 2 x photon angular momentum


FILL
Page 1 of 2
Enter FILL
Loop through 3
3
Initialize $\phi_\pm(\kappa,\lambda) = 0$
Loop on matrix elements to 22
Is photon ℓ > max ℓ
Yes
Eliminates high ℓ terms in reduction loop
22
No
For each matrix element, examine the selection rules $[\Delta(\ell 1 \lambda), \lambda \geq 1, \Delta(j_\kappa j_\kappa{}' \lambda)]$ to index the permissible values of $\lambda - \ell + 2$
Loop on $\lambda - \ell + 2$ to 21
Is $\lambda = \ell + 1$
Yes
18
No
$f = \lambda + 1/2$
Is $\Delta(\ell_{\mp\kappa'} j_\kappa f)$ satisfied
No
17
Yes
Compute R from appropriate algebraic expression (according as $j_\kappa{}' \gtrless \ell_{\mp\kappa'}$)
Multiply by matrix element and C-coefficient and add into $\phi_\pm(\kappa,\lambda)$
17
Is $\lambda = \ell - 1$
Yes
21
No
$f = \lambda - 1/2$
a

FILL
Page 2 of 2


α
Is $\Delta(\ell_{\mp\kappa'}, j_\kappa f)$ satisfied
No
21
Compute contribution to $\phi_{\pm}(\kappa,\lambda)$ as above
21
Continue
22
Continue
Return

```
$IBFTC FILL
      SUBROUTINE FILL (TK,KW,LB,NT,LP,FI)
      COMMON /FAC/FACT(67),RTFAC(95),ROOT(50)
      COMMON /LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)
      DIMENSION TK(200),KW(200),LB(200),FI(30,15)
      LMP=LM+1
      DO 3 LA=1,LMP
      DO 3 K=1,K2M
    3 FI(K,LA)=0.0
      LD=2*LP
      DO 22 N = 1, NT
      L=LB(N)-1
      IF (L.GT.LM)      GO TO 22
      K=KW(N)
      JC=JK(K)
      IF (L.EQ.0)      GO TO 8
      L2=2*L
      JPL=JKB+JC-L2
      JML=IABS (JKB-JC)-L2
      IF (JPL.LT.2)      GO TO 9
    4 IF (L.EQ.1)      GO TO 6
      IF (JML.GT.(-2))      GO TO 6
      NA=1
      NB=3
      GO TO 14
    6 IF (JML.GT.0)      GO TO 8
      NA=2
      NB=3
      GO TO 14
    8 NA=3
      NB=3
      GO TO 14
    9 IF (L.EQ.1)      GO TO 13
      IF (JPL.GE.0)      GO TO 11
      NA=1
      NB=1
      GO TO 14
   11 IF (JML.GT.(-2))      GO TO 13
      NA=1
      NB=2
      GO TO 14
   13 NA=2
      NB=2
   14 DO 21 NL=NA,NB
      LA=L-2+NL
      LAM=2*LA
      IF (NL.EQ.3)      GO TO 18
      LEF=LAM+1
      IF ((LD+JC).LT.LEF)      GO TO 17
      IF (IABS(LD-JC).GT.LEF)      GO TO 17
      IF (JKB.GT.LD)      GO TO 15
      IR=JKB+JC+LAM+4
      IS=JKB-JC+LAM+2
      R=-ROOT(IR)*ROOT(IS)*ROOT(LA)
      GO TO 16
   15 IS=JC+JKB-LAM
      IF (IS.EQ.0)      GO TO 17
      IR=JC-JKB+LAM+2
      R=ROOT(IR)*ROOT(IS)*ROOT(LA)
   16 CALL  COEFS (LEF,LD,JC,1,0,C)
      FI(K,LA)=FI(K,LA)+TK(N)*R*C
```

```
17 IF (NL.EQ.1)       GO TO 21
18 LEF=LAM-1
   IF ((LD+JC).LT.LEF)      GO TO 21
   IF (IABS(LD-JC).GT.LEF)       GO TO 21
   IF (JKB.GT.LD)      GO TO 19
   IS=JC-JKB+LAM
   IF (IS.EQ.0)      GO TO 21
   IR=JC+JKB-LAM+2
   GO TO 20
19 IS=JKB-JC+LAM
   IF (IS.EQ.0)       GO TO 21
   IR=JKB+JC+LAM+2
20 R=ROOT(IR)*ROOT(IS)*ROOT(LA+1)
   CALL  COEFS (LEF,LD,JC,1,0,C)
   FI(K,LA)=FI(K,LA)+TK(N)*R*C
21 CONTINUE
22 CONTINUE
   RETURN
   END
```

SUBROUTINE HUM

Purpose: Computes $H(\varkappa,\mu')$

Method: For every $\varkappa > 0$, the selection rules on the Clebsch-Gordan coefficient are examined to determine the allowed range of λ values. The maximum (positive) value of μ' and its minimum (largest negative) value are computed and indexed for subsequent use. Since $j_\varkappa$ depends only on the absolute value of $\varkappa$, the Clebsch-Gordan coefficient is the same for $(-\varkappa)$ as for $\varkappa$. Over the allowed range of positive μ', the Clebsch-Gordan coefficients are computed for given λ and $\varkappa$, multiplied in turn by $\phi(\varkappa,\lambda)$ and $\phi(-\varkappa,\lambda)$ to obtain corresponding contributions to $H(\varkappa,\mu')$ and $H(-\varkappa,\mu')$, and the products are summed over λ. The results are denoted by HF(K,M) where K is a positive integer. indexing $\varkappa$, and $M = \mu' + 1/2$. The procedure is then repeated for μ' negative, leading to HFM(K,M), where K is as above and $M =-\mu' + 1/2$. The explicit separation of positive and negative μ' terms is useful later on.

Subroutines called: CØEFS, FILL

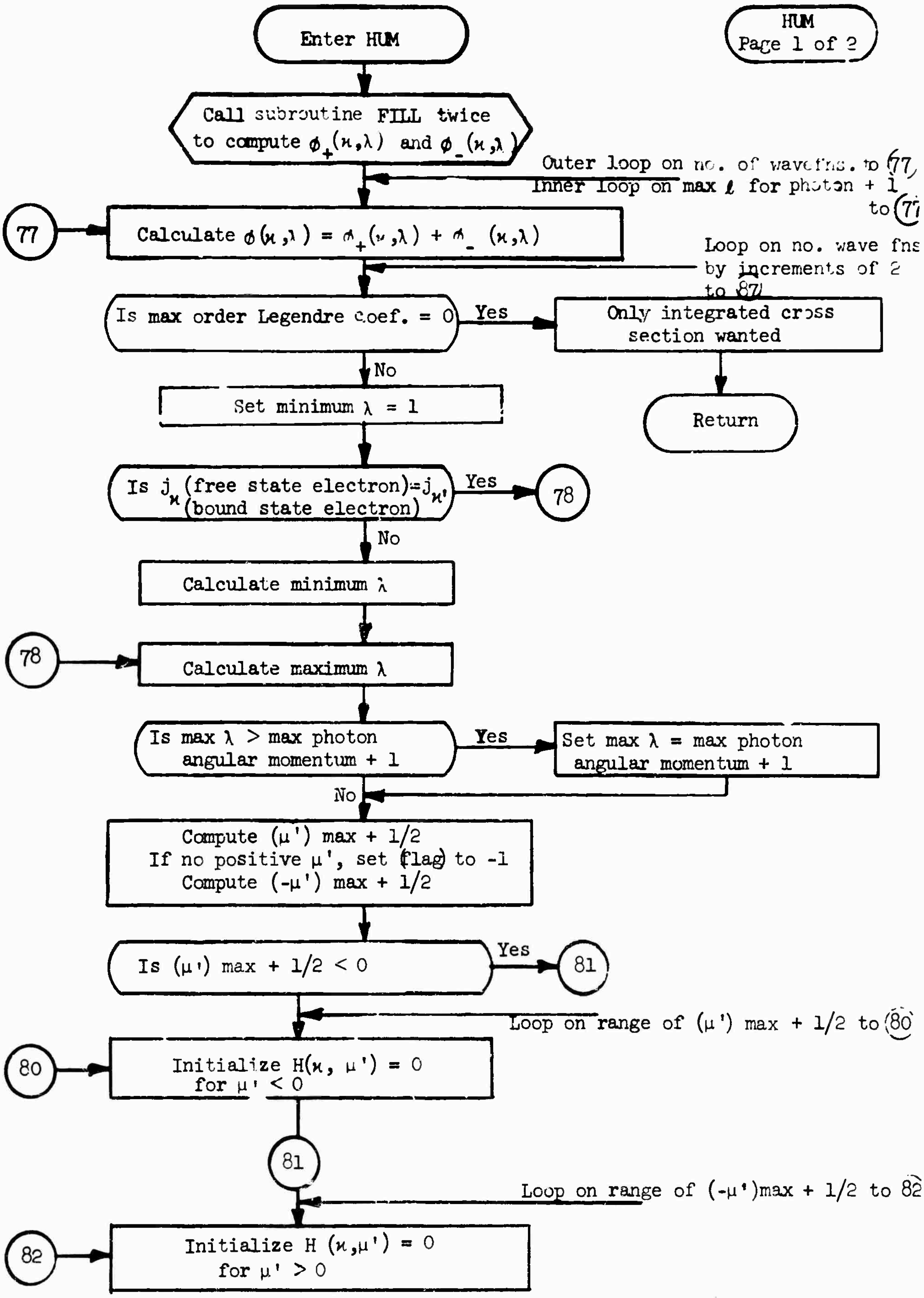
Subroutine called by: RADINT

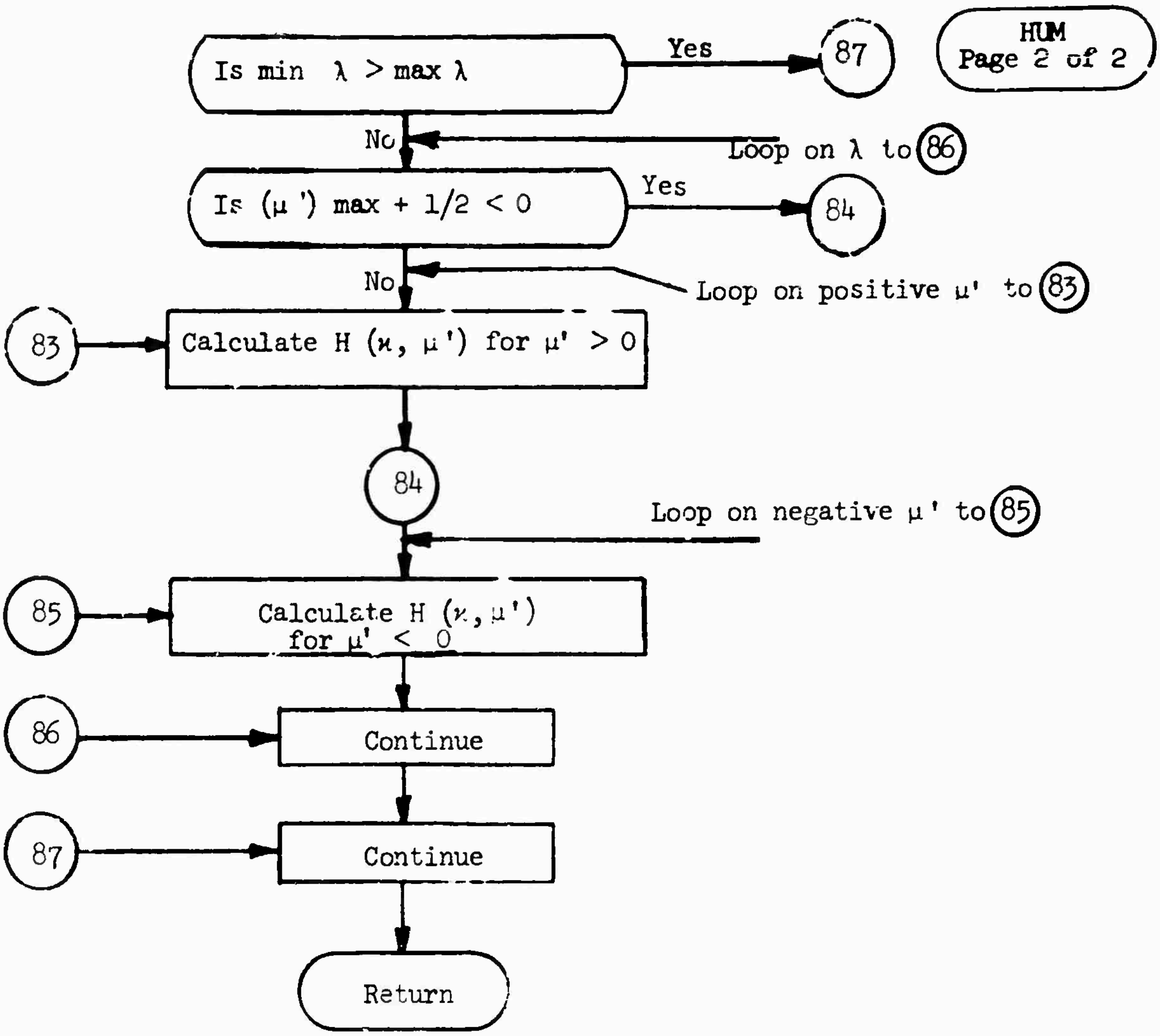
Labelled Common: FIDØ, LIMIT, MAT, QUANT, TRANS, VECT

Local Variables:

Name	Dimension	Mode	Meaning
LMP		I	Maximum photon angular momentum + 1
K		I	Loop index for bound electron state
LA			λ
JC		I	Twice $j_\varkappa$ (free electron state)
LAD		I	Minimum λ from selection rules

Name	Dimension	Mode	Meaning
LAP		I	Maximum λ from selection rules
JPØS		I	$(\mu')_{max} + 1/2$ provided $\mu' > 0$ permitted, -1 otherwise
JNEG		I	$(-\mu')_{max} + 1/2$
M		I	Loop index for μ'
LAM		I	Twice λ
MU		I	Twice μ'


HUM
Page 1 of 2
Enter HUM
Call subroutine FILL twice to compute $\phi_+(\kappa,\lambda)$ and $\phi_-(\kappa,\lambda)$
Outer loop on no. of wavefns. to 77
Inner loop on max ℓ for photon + 1 to 77
77
Calculate $\phi(\kappa,\lambda) = \phi_+(\kappa,\lambda) + \phi_-(\kappa,\lambda)$
Loop on no. wave fns by increments of 2 to 87
Is max order Legendre coef. = 0
Yes
Only integrated cross section wanted
Return
No
Set minimum λ = 1
Is j_κ (free state electron) = $j_{\kappa'}$ (bound state electron)
Yes
78
No
Calculate minimum λ
78
Calculate maximum λ
Is max λ > max photon angular momentum + 1
Yes
Set max λ = max photon angular momentum + 1
No
Compute (μ') max + 1/2
If no positive μ', set flag to -1
Compute $(-\mu')$ max + 1/2
Is (μ') max + 1/2 < 0
Yes
81
Loop on range of (μ') max + 1/2 to 80
80
Initialize $H(\kappa, \mu') = 0$ for $\mu' < 0$
81
Loop on range of $(-\mu')$max + 1/2 to 82
82
Initialize $H(\kappa,\mu') = 0$ for $\mu' > 0$


HUM
Page 2 of 2
Is min λ > max λ
Yes
87
No
Loop on λ to 86
Is (μ ') max + 1/2 < 0
Yes
84
No
Loop on positive μ' to 83
83
Calculate H (ϰ, μ') for μ' > 0
84
Loop on negative μ' to 85
85
Calculate H (ϰ, μ') for μ' < 0
86
Continue
87
Continue
Return

```
$IBFTC HUMM
      SUBROUTINE HUM                                                    HUM00010
      COMMON/FIDO/FI(30,15),D(30),JMP,NAME,SHELL,QV,E3,IZ               HUM00020
      COMMON /LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB          HUM00030
      COMMON/MAT/SF(30),SG(30),FK(200),FKP(200),SFK(200),SFKP(200),RCUT HUM00040
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)  HUM00050
      COMMON/TRANS/HF(30,15),HFM(30,15),JNG(30),JPS(30)                 HUM00060
      COMMON/VECT/KF(200),KG(200),LBES(200),LBS(200),LKB                HUM00070
      DIMENSION FT(30,15),FTP(30,15)                                    HUM00080
      LMP=LM+1                                                          HUM00090
      CALL FILL (FK,KG,LBES,NK,LMKB,FT)                                 HUM00100
      CALL FILL (FKP,KF,LBS,NKP,LKB,FTP)                                HUM00110
      DO 77 K=1,K2M                                                     HUM00120
      DO 77 LA=1,LMP                                                    HUM00130
   77 FI(K,LA)=FT(K,LA)+FTP(K,LA)                                       HUM00140
      IF ( JM .EQ. 0 )                    RETURN                        HUM00145
      DO 87 K=1,K2M,2                                                   HUM00150
      KP = K + 1                                                        HUM00160
      JC=JK(K)                                                          HUM00170
      LAD=1                                                             HUM00180
      IF (JC.EQ.JKB)      GO TO 78                                      HUM00190
      LAD=IABS (JC-JKB)/2                                               HUM00200
   78 LAP = (JC+JKB)/2                                                  HUM00210
      IF ( LAP .GT. LMP )          LAP = LMP                            HUM00220
      JPOS=JC-2                                                         HUM00230
      IF (JC.LT.4)      GO TO 79                                        HUM00240
      JPOS=MIN0 (JKB,JPOS)                                              HUM00250
      JPOS=(JPOS+1)/2                                                   HUM00260
   79 JNEG=MIN0 (JKB,(JC+2))                                            HUM00270
      JNEG=(JNEG+1)/2                                                   HUM00280
      JNG(K)=JNEG                                                       HUM00290
      JPS(K)=JPOS                                                       HUM00300
      JNG(KP)=JNEG                                                      HUM00310
      JPS(KP)=JPOS                                                      HUM00320
      IF (JPOS.LT.0)      GO TO 81                                      HUM00330
      DO 80 M=1,JPOS                                                    HUM00340
      HF(K,M)  =  0.0                                                   HUM00350
   80 HF(KP,M) = 0.0                                                    HUM00360
   81 DO 82 M=1,JNEG                                                    HUM00370
      HFM(K,M) = 0.0                                                    HUM00380
   82 HFM(KP,M) = 0.0                                                   HUM00390
      IF (LAD.GT.LAP)      GO TO 87                                     HUM00400
      DO 86 LA=LAD,LAP                                                  HUM00410
      LAM=2*LA                                                          HUM00420
      IF (JPOS.LT.0)      GO TO 84                                      HUM00430
      DO 83 M=1,JPOS                                                    HUM00440
      MU=2*M-1                                                          HUM00450
      CALL COEFS (LAM,JKB,JC,2,MU,C)                                    HUM00460
      HF(K,M) = HF(K,M) + FI(K,LA) * C                                  HUM00470
   83 HF(KP,M) = HF(KP,M) + FI(KP,LA) * C                               HUM00480
   84 DO 85 M=1,JNEG                                                    HUM00490
      MU=1-2*M                                                          HUM00500
      CALL COEFS (LAM,JKB,JC,2,MU,C)                                    HUM00510
      HFM(K,M) = HFM(K,M) + FI(K,LA) * C                                HUM00520
   85 HFM(KP,M) = HFM(KP,M) + FI(KP,LA) * C                             HUM00530
   86 CONTINUE                                                          HUM00540
   87 CONTINUE                                                          HUM00550
      RETURN                                                            HUM00560
      END                                                               HUM00570
```

This page is intentionally left blank.

SUBROUTINE INTERP

Purpose: Interpolates on the bound-state tabulations for radii greater than one-half Bohr unit where the integration grid is much finer than the table, to obtain intermediate values of the bound-state wavefunctions and corresponding potential.

Method: Linear interpolation between successive entries in table.

Subroutine called: None

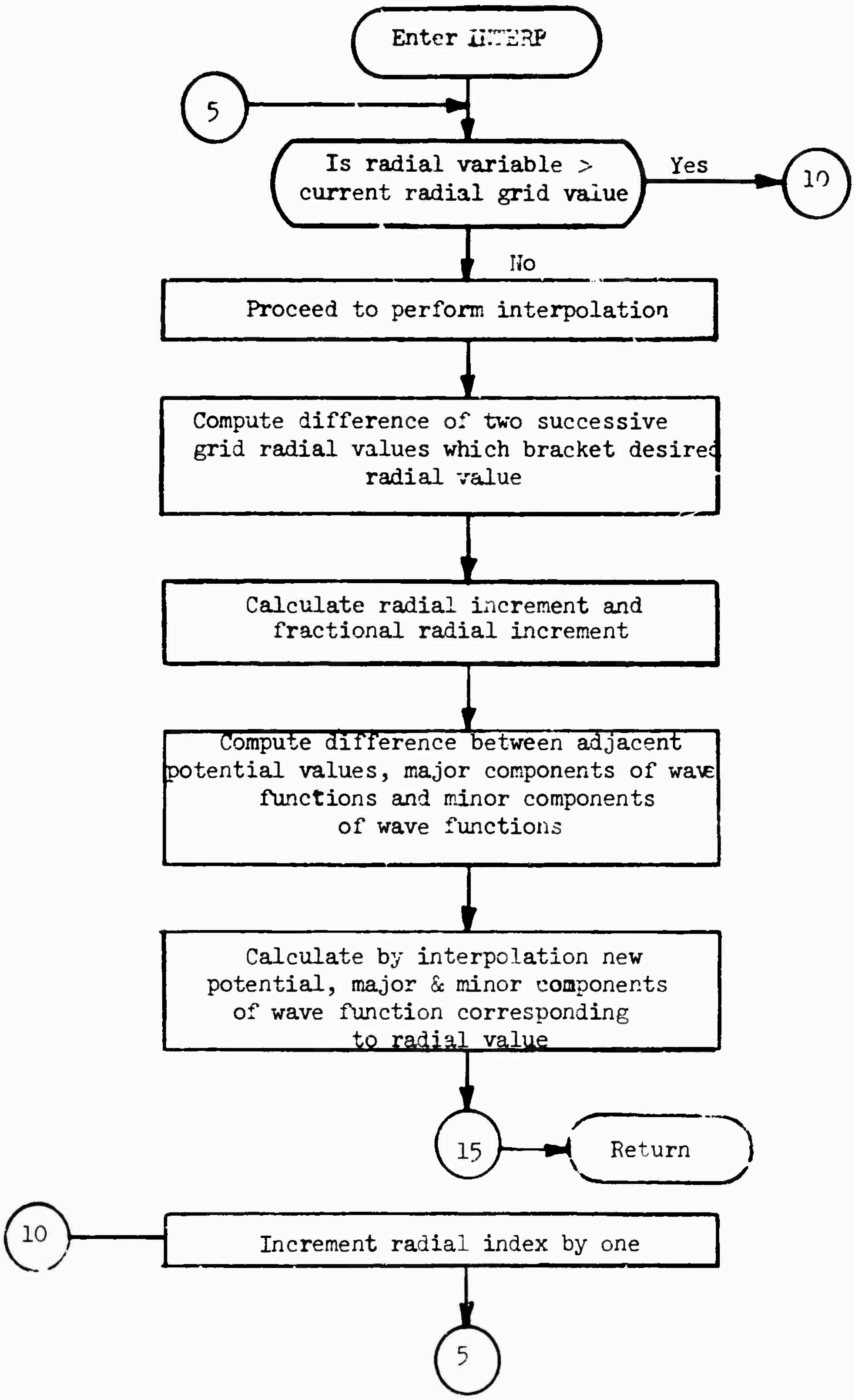
Subroutine called by: XDERIV

Labelled Common: LIMIT, ØNWARD, TAPES

Local Variables:

Name	Dimension	Mode	Meaning
DX		R	Difference between two successive tabulated radial values
DL		R	Radial increment; difference between integrating radius and lower grid point
QUØT		R	Fractional radial increment
DV		R	Difference between two successive tabulated potential values
DGB		R	Difference between two successive tabulated values of "large" component of wavefunction.
DFB		R	Difference between two successive tabulated values of "small" component of wavefunction.

INTERP
Page 1 of 1


Enter INTERP
5
Is radial variable > current radial grid value
Yes
10
No
Proceed to perform interpolation
Compute difference of two successive grid radial values which bracket desired radial value
Calculate radial increment and fractional radial increment
Compute difference between adjacent potential values, major components of wave functions and minor components of wave functions
Calculate by interpolation new potential, major & minor components of wave function corresponding to radial value
15
Return
10
Increment radial index by one
5

```
$IBFTC INTP
      SUBROUTINE INTERP                                                  INT00010
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB            INT00020
      COMMON /ONWARD/RX,SCX,GBX,FBX                                      INT00030
      COMMON/TAPES/X(1500),SCF(1500),FB(1500),GB(1500),GAMB,SCREEN       INT00040
    5 IF (RX.GT.X(NTAB+1))      GO TO 10                                 INT00050
      DX = X(NTAB+1)-X(NTAB)                                             INT00060
      DL = RX-X(NTAB)                                                    INT00070
      QUOT = DL/DX                                                       INT00080
      DV = SCF(NTAB+1)-SCF(NTAB)                                         INT00090
      DGB = GB(NTAB+1)-GB(NTAB)                                          INT00100
      DFB = FB(NTAB+1)-FB(NTAB)                                          INT00110
      SCX = SCF(NTAB)+QUOT*DV                                            INT00120
      GBX = GB(NTAB)+QUOT*DGB                                            INT00130
      FBX = FB(NTAB)+QUOT*DFB                                            INT00140
      GO TO 15                                                           INT00150
   10 NTAB = NTAB+1                                                      INT00160
      GO TO 5                                                            INT00170
   15 RETURN                                                             INT00180
      END                                                                INT00190
```

This page is intentionally left blank.

SUBROUTINE LEGEND

Purpose: Computes the Legendre coefficients of the cross section.

Method: For j = 0, this consists of carrying out the sum

$$D_0 = \frac{\pi^2 e^2 w}{2\epsilon} \sum_\lambda \frac{1}{2\lambda + 1} \sum_\kappa \phi^2(\kappa,\lambda) \text{ over all } \kappa \text{ and } \lambda$$

(The ϕ's have been initialized.) For $j > 0$, subroutine MUSS is called where the $T_j(\kappa,\kappa'')$ of the sum

$$D_j = \frac{\pi^2 e^2 w}{2\epsilon} (-1)^j \sum_{\kappa\kappa''} \cos(\delta_\kappa - \delta_{\kappa''}) \, T_j(\kappa,\kappa'') \text{ is performed.}$$

The summation $((-1)^j \sum_{\kappa\kappa''} \cos( )T_j ( ))$ is subject to the selection rules $\Delta(j_\kappa j_{\kappa''},j)$, $\Delta(\ell_\kappa \ell_{\kappa''},j)$ and $\ell_\kappa + \ell_{\kappa''} + j$ = even integer. The diagonal terms in the double sum are done first (with the cosine equal to unity), carried over $j_\kappa > 1/2$ and contributing to even j terms only. Since the off-diagonal terms are symmetric in κ and κ'', twice the sum with $K'' < K$ is taken. For given κ and κ'', the smallest j for which there can be a contribution is $|\ell_\kappa - \ell_{\kappa''}|$ provided $|j_\kappa - j_{\kappa''}|$ is not larger; otherwise $j = |j_\kappa - j_{\kappa''}|$ has the wrong parity and the minimum j value is $|j_\kappa - j_{\kappa''}| + 1 = |\ell_\kappa - \ell_{\kappa''}| + 2$. There may also be contributions for larger j (going up in steps of two to preserve parity) up to the lesser of $(\ell_\kappa + \ell_{\kappa''})$ and $(j_\kappa + j_{\kappa''})$, or up to an assigned maximum j if smaller.

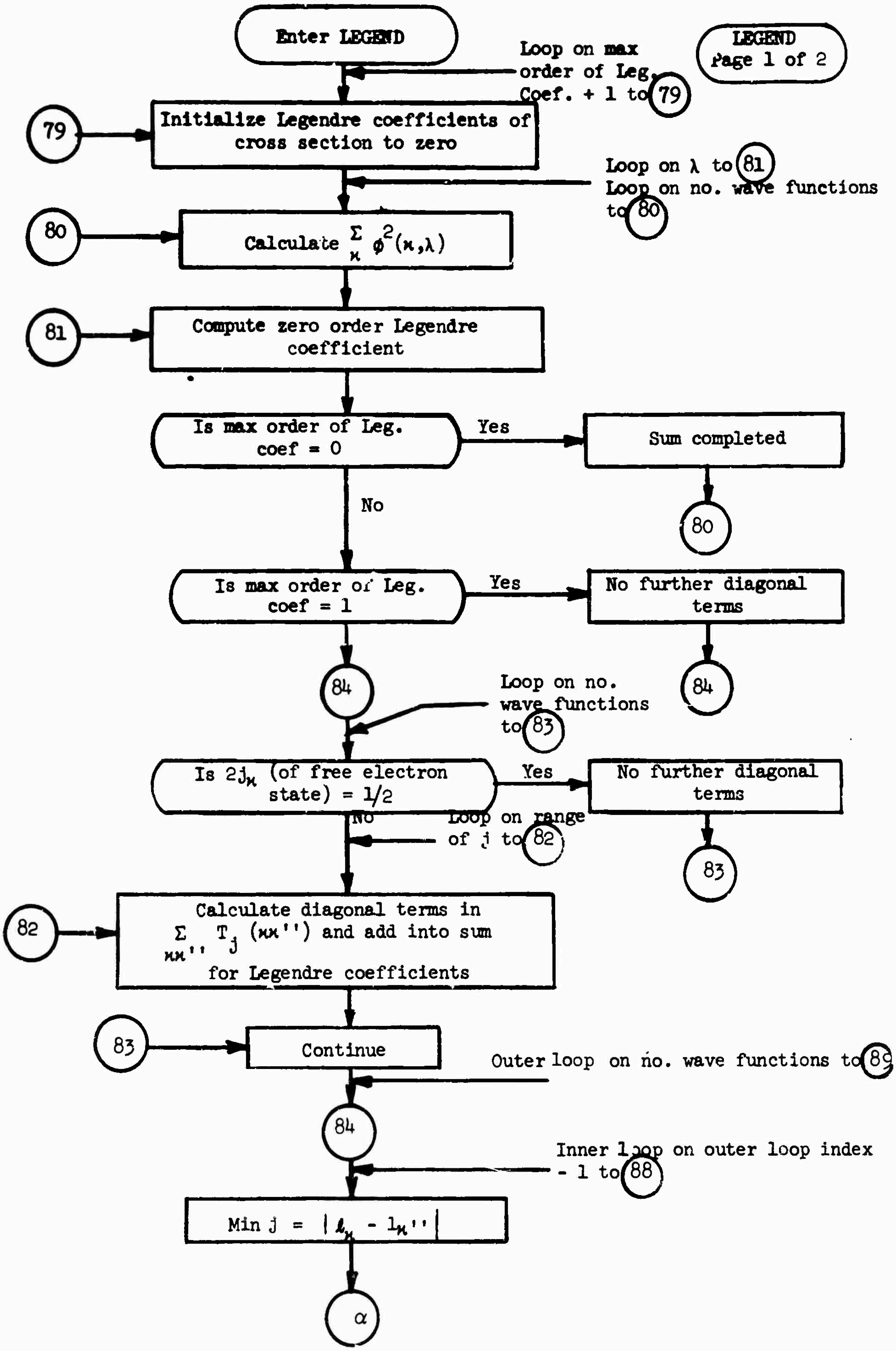
Subroutine called: MUSS

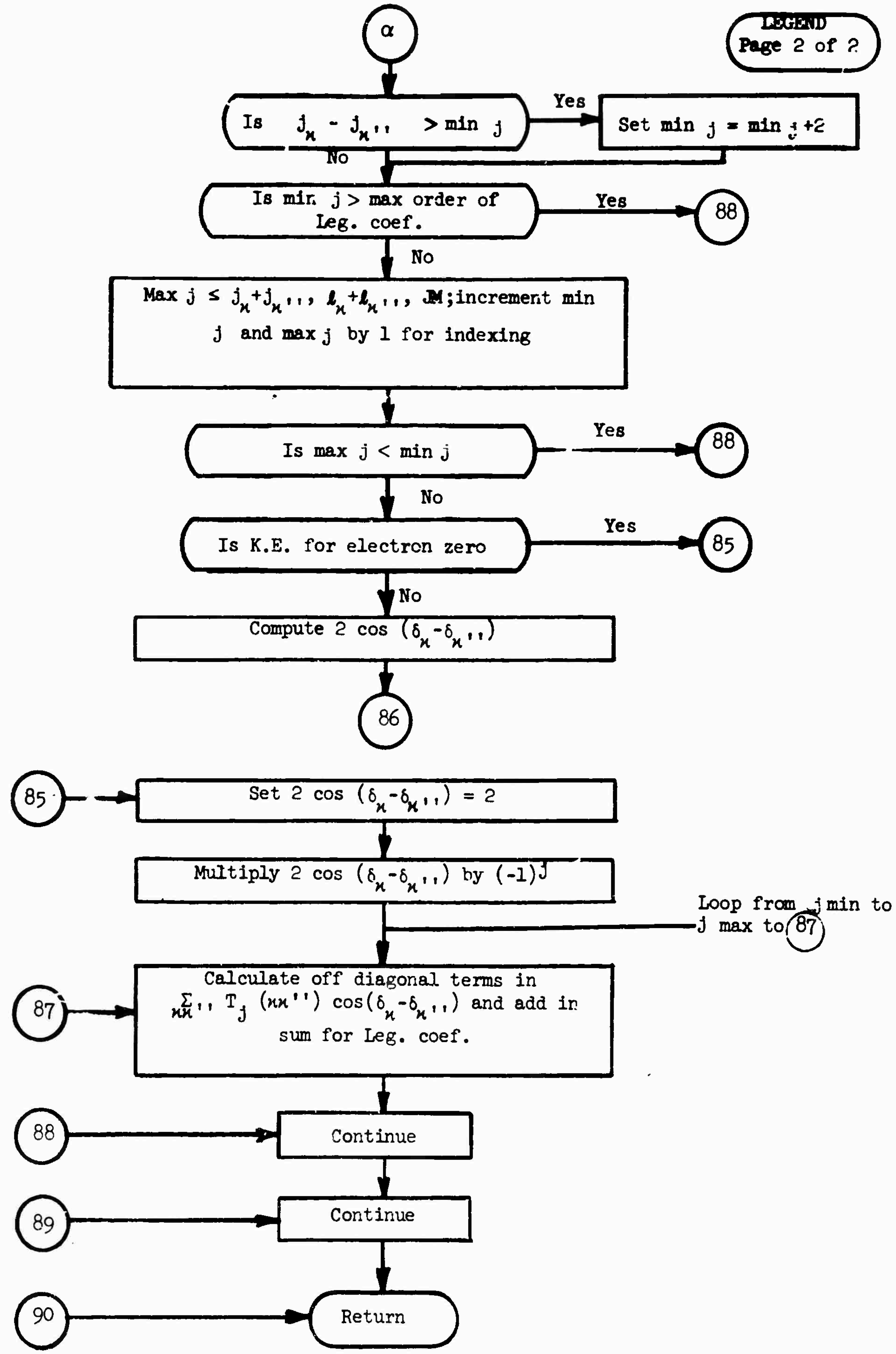
Subroutine called by: PELEC

Labelled Common: FIDØ, LIMIT, QUANT

Local Variables:

Name	Dimension	Mode	Meaning
LMP		I	Max photon angular momentum + 1
J		I	Order of Legendre coefficient
LA		I	λ
STAT		R	$2\lambda + 1$
SKP		R	$\sum_{\kappa} \phi^2(\kappa,\lambda)$
JMAX		I	Max j contribution for given κ and κ''
JMIN		I	Min j contribution for given κ and κ''
K		I	Loop index for bound electron state
TJ		R	$T_j (\kappa,\kappa'')$
JDIF		I	$\left\| \ell_\kappa - \ell_{\kappa''} \right\|$
CØD		R	$2 \cos (\delta_\kappa - \delta_{\kappa''})$
MINI		I	$(-1)^{j-1}$


Enter LEGEND
LEGEND Page 1 of 2
Loop on max order of Leg. Coef. + 1 to 79
79
Initialize Legendre coefficients of cross section to zero
Loop on λ to 81
Loop on no. wave functions to 80
80
Calculate $\sum_{\kappa} \phi^2(\kappa,\lambda)$
81
Compute zero order Legendre coefficient
Is max order of Leg. coef = 0
Yes
Sum completed
80
No
Is max order of Leg. coef = 1
Yes
No further diagonal terms
84
84
Loop on no. wave functions to 83
Is $2j_\kappa$ (of free electron state) = 1/2
Yes
No further diagonal terms
83
No
Loop on range of j to 82
82
Calculate diagonal terms in $\sum_{\kappa\kappa''} T_j(\kappa\kappa'')$ and add into sum for Legendre coefficients
83
Continue
Outer loop on no. wave functions to 89
84
Inner loop on outer loop index - 1 to 88
Min j = $|l_\kappa - l_{\kappa''}|$
α


LEGEND
Page 2 of 2
α
Is $j_{\varkappa} - j_{\varkappa''}$ > min j
Yes
Set min j = min j +2
No
Is min j > max order of Leg. coef.
Yes
88
No
Max j ≤ $j_{\varkappa}+j_{\varkappa''}$, $\ell_{\varkappa}+\ell_{\varkappa''}$, JM;increment min j and max j by 1 for indexing
Is max j < min j
Yes
88
No
Is K.E. for electron zero
Yes
85
No
Compute 2 cos $(\delta_{\varkappa}-\delta_{\varkappa''})$
86
85
Set 2 cos $(\delta_{\varkappa}-\delta_{\varkappa''})$ = 2
Multiply 2 cos $(\delta_{\varkappa}-\delta_{\varkappa''})$ by $(-1)^j$
Loop from j min to j max to 87
87
Calculate off diagonal terms in $\sum_{\varkappa\varkappa''} T_j(\varkappa\varkappa'') \cos(\delta_{\varkappa}-\delta_{\varkappa''})$ and add in sum for Leg. coef.
88
Continue
89
Continue
90
Return

```
$IBFTC LEGE
      SUBROUTINE LEGEND                                                 LEG00010
      COMMON/FIDO/FI(30,15),D(30),JMP,NAME,SHELL,QV,EB,IZ               LEG00020
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB           LEG00030
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)  LEG00040
      LMP=LM+1                                                          LEG00050
      DO 79 J=1,JMP                                                     LEG00060
   79 D(J)=0.0                                                          LEG00070
      DO 81 LA=1,LMP                                                    LEG00080
      STAT=2*LA+1                                                       LEG00090
      SKP=0.0                                                           LEG00100
      DO 80 K=1,K2M                                                     LEG00110
   80 SKP = SKP+FI(K,LA)*FI(K,LA)                                       LEG00120
   81 D(1)=D(1)+SKP/STAT                                                LEG00130
      IF (JM.EQ.0)      GO TO 90                                        LEG00140
      IF (JM.EQ.1)      GO TO 84                                        LEG00150
      DO 83 K=1,K2M                                                     LEG00160
      IF ( JK(K) .EQ. 1 )          GO TO 83                             LEG00170
      JMAX=MIN0(JK(K),2*LK(K),JM)+1                                     LEG00180
      DO 82 J=3,JMAX,2                                                  LEG00190
      CALL MUSS(K,K,J,TJ)                                               LEG00200
   82 D(J)=D(J)+TJ                                                      LEG00210
   83 CONTINUE                                                          LEG00220
   84 DO 89 K=2,K2M                                                     LEG00230
      KLS=K-1                                                           LEG00240
      DO 88 KK=1,KLS                                                    LEG00250
      JMIN=IABS(LK(K)-LK(KK))                                           LEG00260
      JDIF=IABS(JK(K)-JK(KK))                                           LEG00270
      IF (JDIF.GT.(2*JMIN))      JMIN=JMIN+2                            LEG00280
      IF (JMIN.GT.JM)      GO TO 38                                     LEG00290
      JMIN=JMIN+1                                                       LEG00300
      JMAX=MIN0(((JK(K)+JK(KK))/2),(LK(K)+LK(KK)),JM)+1                 LEG00310
      IF ( JMAX .LT. JMIN )                  GO TO 88                   LEG00320
      IF ( IEND .EQ. 1 )      GO TO 85                                  LEG00330
      COD = 2.0 * ( CR(K)*CR(KK) + SI(K)*SI(KK) )                       LEG00340
      GO TO 86                                                          LEG00350
   85 COD = 2.0                                                         LEG00360
   86 MINI= MOD(JMIN,2)                                                 LEG00370
      IF (MINI.EQ.0)      COD=-COD                                      LEG00380
      DO 87 J=JMIN,JMAX,2                                               LEG00390
      CALL MUSS (K,KK,J,TJ)                                             LEG00400
   87 D(J)=D(J)+TJ*COD                                                  LEG00410
   88 CONTINUE                                                          LEG00420
   89 CONTINUE                                                          LEG00430
   90 RETURN                                                            LEG00440
      END                                                               LEG00450
```

This page is intentionally left blank.

SUBROUTINE LØGGAM

Purpose: Computes the natural logarithm of the gamma function for complex arguments, i.e. Re $\ln \Gamma(x + iy)$ and Im $\ln \Gamma(x + iy)$

Method: a) Set $\Gamma(z) = -z + (z - 1/2) \ln z + \ln\sqrt{2\pi} + J(z)$ where $J(z)$ is given as a continuous fraction. See Wall, "Analytic Theory of Continued Functions," p. 364, formula 93.9.

b) For $x > 2$, $\ln \Gamma(z)$ is computed from the recursion relation:

$$\ln\Gamma(z) = \ln\Gamma(1 + z) - \ln z$$

c) For negative x, the Im $\ln\Gamma(z)$ can be thought of as being equal to $V + 2\pi k$ where k is an integer and V is given by this routine.

Restrictions: a) x and y may not both be equal to zero.

b) if $y = 0$, x may not be equal to a negative integer.

This routine is taken from M. S. Shapiro and M. Goldstein, "A Collection of Mathematical Computer Routines," NYO-1480-14 (1965).

Subroutine called: None

Subroutine called by: RADINT

Argument sequence: (X, Y, U, V)

Argument List:

Name	Dimension	Mode	Meaning
X		R	Real part of argument
Y		R	Imaginary part of argument
U		R	Real part of result
V		R	Imaginary part of result

```
$IBFTC LGAM
      SUBROUTINE LOGGAM(X,Y,U,V)
C THIS SUBROUTINE COMPUTES THE NATURAL LOG OF THE GAMMA FUNCTION FOR
C COMPLEX ARGUMENTS.  THE ROUTINE IS ENTERED BY THE STATEMENT
C     CALL LOGGAM(X,Y,U,V)
C WHERE    X IS THE REAL PART OF THE ARGUMENT
C          Y IS THE IMAGINARY PART OF THE ARGUMENT
C          U IS THE REAL PART OF THE RESULT
C          V IS THE IMAGINARY PART OF THE RESULT
      DIMENSION H(7)
      H(1)=2.269488974
      H(2)=1.517473649
      H(3)=1.011523068
      H(4)=.5256064690
      H(5)=.2523809524
      H(6)=0.0333333333
      H(7)=0.0833333333
      E2=1.57079632679
      E8=3.14159265359
      B1=0.0
      B2=0.0
      J=2
      X2=X
    4 IF(X)1,2,3
    3 B6=ATAN (Y/X)
      T=X*X
    5 B7=Y*Y+T
C  REAL PART OF LOG
      T1=0.5*ALOG(B7)
      IF(X-2.0)7,7,6
    7 B1=B1+B6
      B2=B2+T1
      X=X+1.0
      J=1
      GO TO 4
    6  T3=-Y*B6+(T1*(X-.5)-X+.9189365332 )
      T2=B6*(X-.5)+Y*T1-Y
      T4=X
      T5=-Y
      T1=B7
      DO 8 I=1,7
      T=H(I)/T1
      T4=T*T4+X
      T5=-(T*T5+Y)
    8 T1=T4*T4+T5*T5
      T3=T4-X+T3
      T2=-T5-Y+T2
      GO TO (9,10),J
    9 T3=T3-B2
      T2=T2-B1
   10 IF(X2)11,12,12
   12 U=T3
      V=T2
      X=X2
      RETURN
   11 U=T3-E4
      V=T2-E5
      X=X2
      RETURN
C    X IS ZERO
    2 T=0.0
      IF(Y)13,14,15
```

```
13 B6=-E2                                                              LGM00620
   GO TO 5                                                             LGM00630
15 B6=E2                                                               LGM00640
   GO TO 5                                                             LGM00650
C     X IS NEGATIVE                                                    LGM00660
 1 E4=0.0                                                              LGM00670
   E5=0.0                                                              LGM00680
   IE6=0                                                               LGM00690
16 E4=E4+.5*(ALOG(X*X +Y*Y ))                                          LGM00700
   E5=E5+ATAN (Y/X)                                                    LGM00710
   IE6=IE6+1                                                           LGM00720
   X=X+1.0                                                             LGM00730
   IF(X)16,17,17                                                       LGM00740
17 IF( MOD (IE6,2))18,4,18                                             LGM00750
18 E5=E5+E8                                                            LGM00760
   GO TO 4                                                             LGM00770
14 WRITE(6,19)              X2,Y                                       LGM00780
19 FORMAT(29H ATTEMPTED TO TAKE LOGGAM OF 2HX=F6.0,1X2HY=F6.0)         LGM00790
   CALL EXIT                                                           LGM00800
   RETURN                                                              LGM00810
   END                                                                 LGM00820
```

This page is intentionally left blank.

SUBROUTINE MUSS

Purpose: Performs the μ' sum and computes the term $T_j\,(\kappa,\kappa'') =$

$$C(j_{\kappa''}, j_{\kappa} j;\ 1/2, -1/2) \sum_{\mu'} (-1)^{\mu' + 1/2}\ H(\kappa,\mu')\ H(\kappa'',\mu')$$

$C(j_{\kappa''}, j_{\kappa} j;\ \mu' + 1, -\mu' -1)$ which is used in computation of D_j (Legendre coefficient).

Method: The sum is carried out first for $(\mu' + 1/2) \geq 0$, computing the C-coefficients and recording them as an indexed variable, the upper limit on μ' being the lesser of the upper limits for κ and κ'' computed in subroutine HUM. The sum for the negative $(\mu' + 1/2)$ is similarly carried up to the lesser of the maximum $|\mu'|$ values for negative μ', but the C-coefficient is obtained from the previously obtained ones by a symmetry operation (introducing at most a sign change). The C-coefficient in front of the sum is the one computed for $\mu' = -1/2$.

Subroutine called: CØEFS

Subroutine called by: LEGEND

Labelled Common: QUANT, TRANS

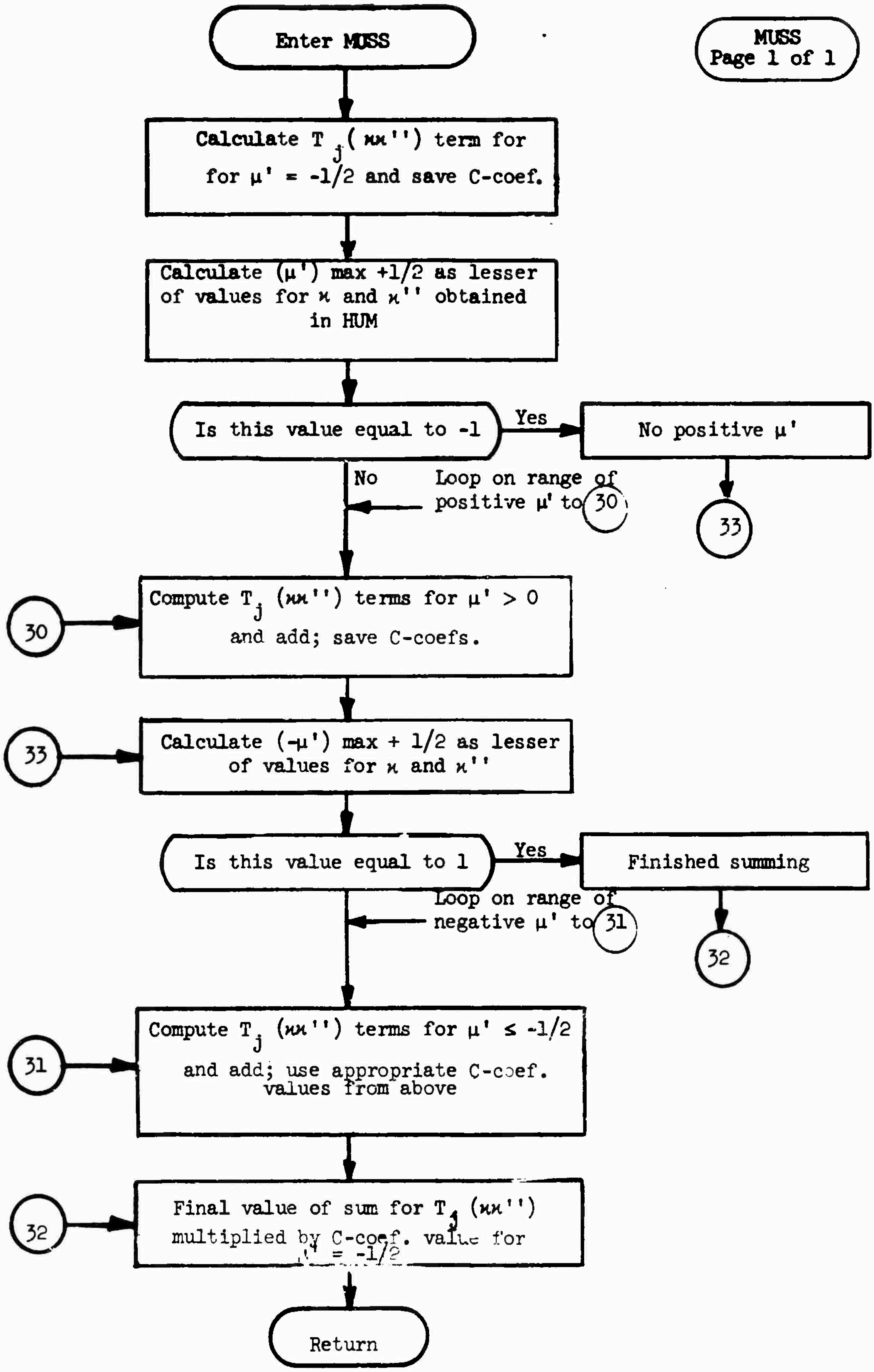
Argument sequence: (K, KK, J, TJ)

Argument List:

Name	Dimension	Mode	Meaning
K		I	Index for κ
KK		I	Index for κ''
J		I	Order of Legendre coefficient + 1
TJ		R	$T_j\,(\kappa,\kappa'')$

Local Variables:

Name	Dimension	Mode	Meaning
JD		I	Twice order of Legendre coefficient
C		R	Clebsch-Gordan coefficient
CL	30	R	Clebsch-Gordan coefficient saved
JPØS		I	Range of positive μ' (-1 means none)
M		I	Loop index for μ'
PM		R	$(-1)^{\mu' + 1/2}$ (times $i^{j_\kappa + j_{\kappa''} + j}$ if μ' negative)
MP		I	$2 \mid \mu' + 1 \mid$
JNEG		I	Range of negative μ'
JMØD		I	$i^{j_\kappa + j_{\kappa''} + j}$


Enter MUSS
MUSS Page 1 of 1
Calculate T_j (κκ'') term for for μ' = -1/2 and save C-coef.
Calculate (μ') max +1/2 as lesser of values for κ and κ'' obtained in HUM
Is this value equal to -1
Yes
No positive μ'
33
No
Loop on range of positive μ' to 30
30
Compute T_j (κκ'') terms for μ' > 0 and add; save C-coefs.
33
Calculate (-μ') max + 1/2 as lesser of values for κ and κ''
Is this value equal to 1
Yes
Finished summing
32
Loop on range of negative μ' to 31
31
Compute T_j (κκ'') terms for μ' ≤ -1/2 and add; use appropriate C-coef. values from above
32
Final value of sum for T_j (κκ'') multiplied by C-coef. value for μ' = -1/2
Return

```
$IBFTC MUSE
      SUBROUTINE MUSS (K,KK,J,TJ)
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)
      COMMON/TRANS/HF(30,15),HFM(30,15),JNG(30),JPS(30)
      DIMENSION CL(30)
      JD=2*(J-1)
      CALL  COEFS(JK(K),JK(KK),JD,1,-1,C)
      TJ = HFM(K,1) * HFM(KK,1) * C
      CL(1)=C
      JPOS=MIN0(JPS(K),JPS(KK))
      IF (JPOS.EQ.(-1))      GO TO 33
      PM=1.0
      DO 30 M=1,JPOS
      PM=-PM
      MP=2*M+1
      CALL  COEFS(JK(K),JK(KK),JD,MP,-MP,C)
      CL(MP)=C
   30 TJ = TJ + HF(K,M) * HF(KK, M) * C * PM
   33 JNEG=MIN0(JNG(K),JNG(KK))
      IF (JNEG.EQ.1)      GO TO 32
      JMOD=MOD((JK(K)+JK(KK)+JD),4)
      PM=-1.0
      IF (JMOD.EQ.0)      PM=1.0
      DO 31 M=2,JNEG
      PM=-PM
      MP=2*M-3
   31 TJ = TJ + HFM(K,M) * HFM(KK,M) * CL(MP) * PM
   32 TJ=TJ*CL(1)
      RETURN
      END
```

SUBROUTINE RADINT

Purpose: This is the control subroutine for the integrations.

Method: The requisite coefficients and the initial values of the free-electron wavefunctions and their derivatives are computed. The radial integrals are performed by calling the Runge-Kutta integration subroutines in DØ loops. Normalization factors are obtained from WNORM and applied to the matrix elements. The phase shifts are obtained by wave-matching. Finally, subroutine HUM is called to start the angular momentum sums.

Subroutines called: LØGGAM, RKUT, DERIV, XDERIV, XRKUT, WNØRM, SPHBES, HUM

Subroutine called by: PELEC

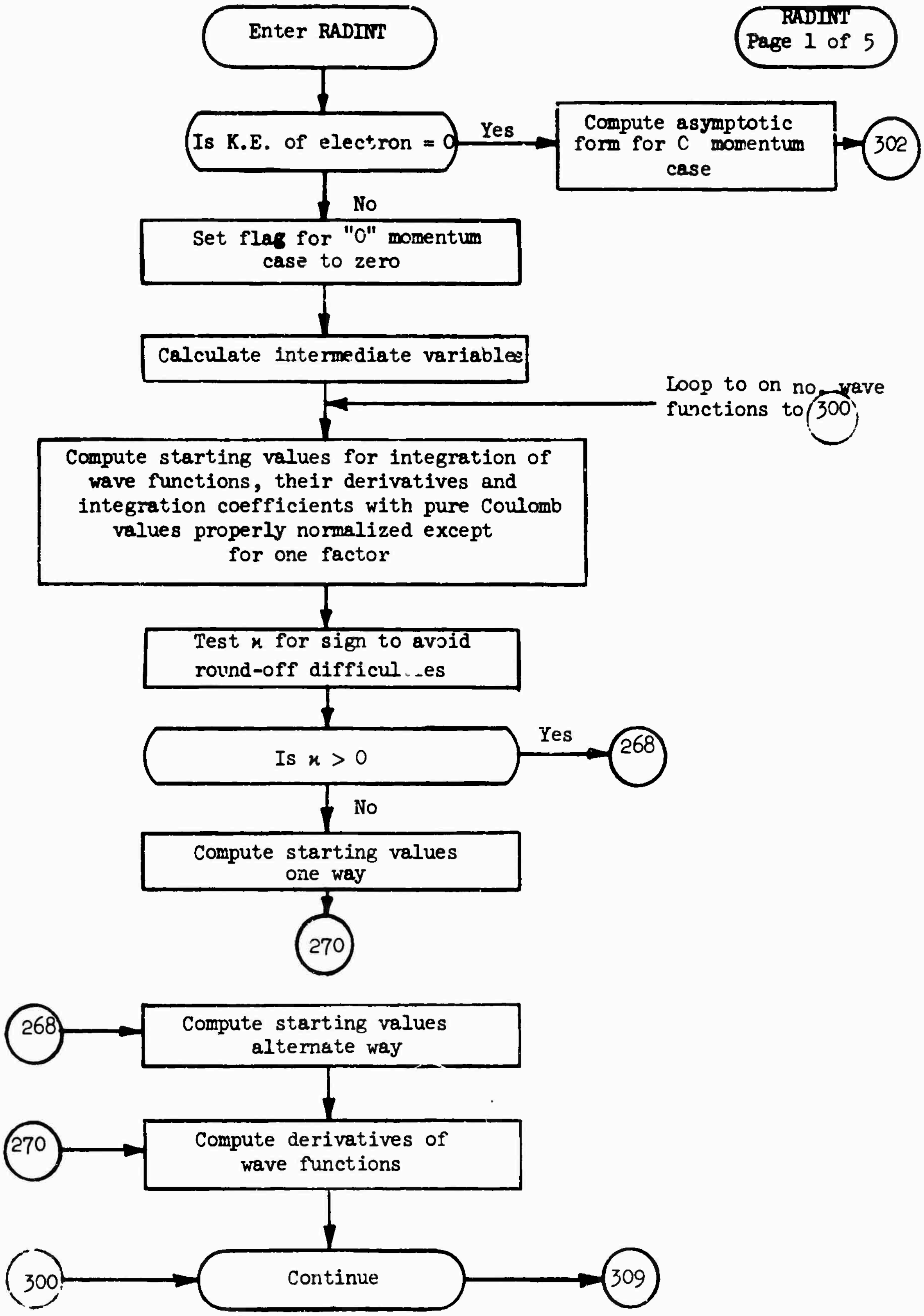
Variables in unlabelled Common: PI, HALFPI, FØURPI, RAD, SQ2, Q, ZA, ZAZA,
EFN, EGN, V, CG, GAM

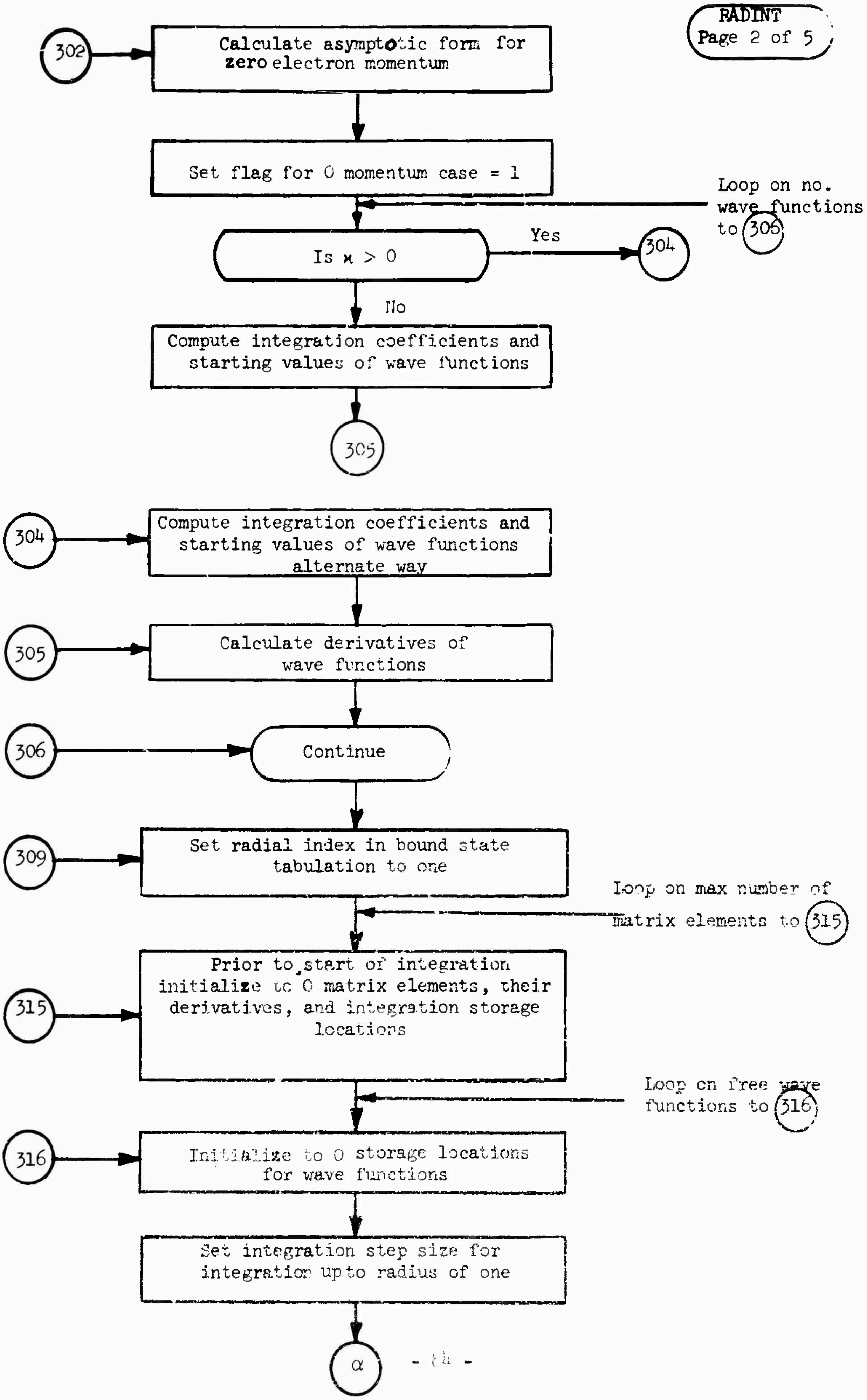
Labelled Common: BESSEL, DFUNC, LIMIT, MAT, QUANT, VECT, ØNWARD

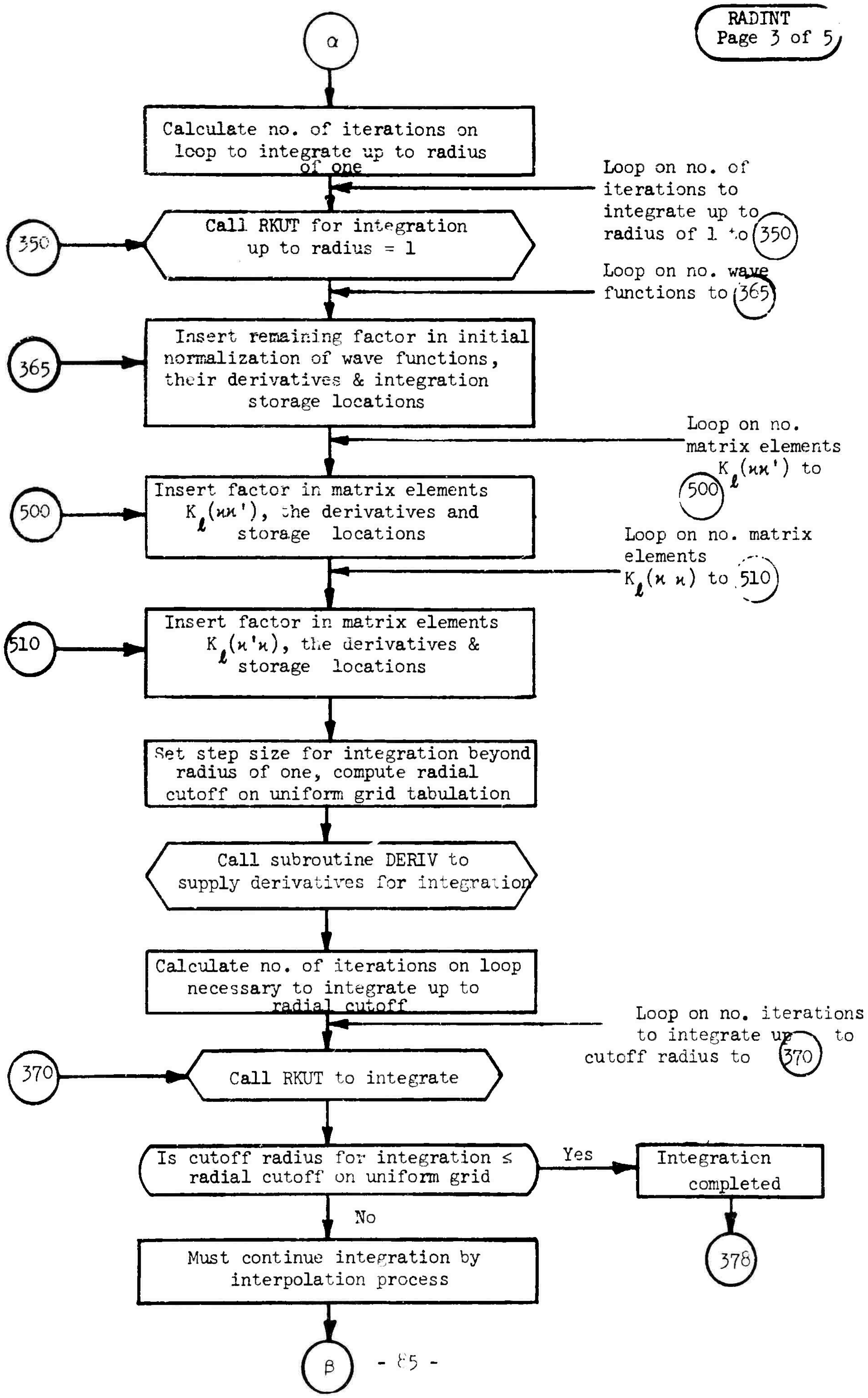
Local Variables:

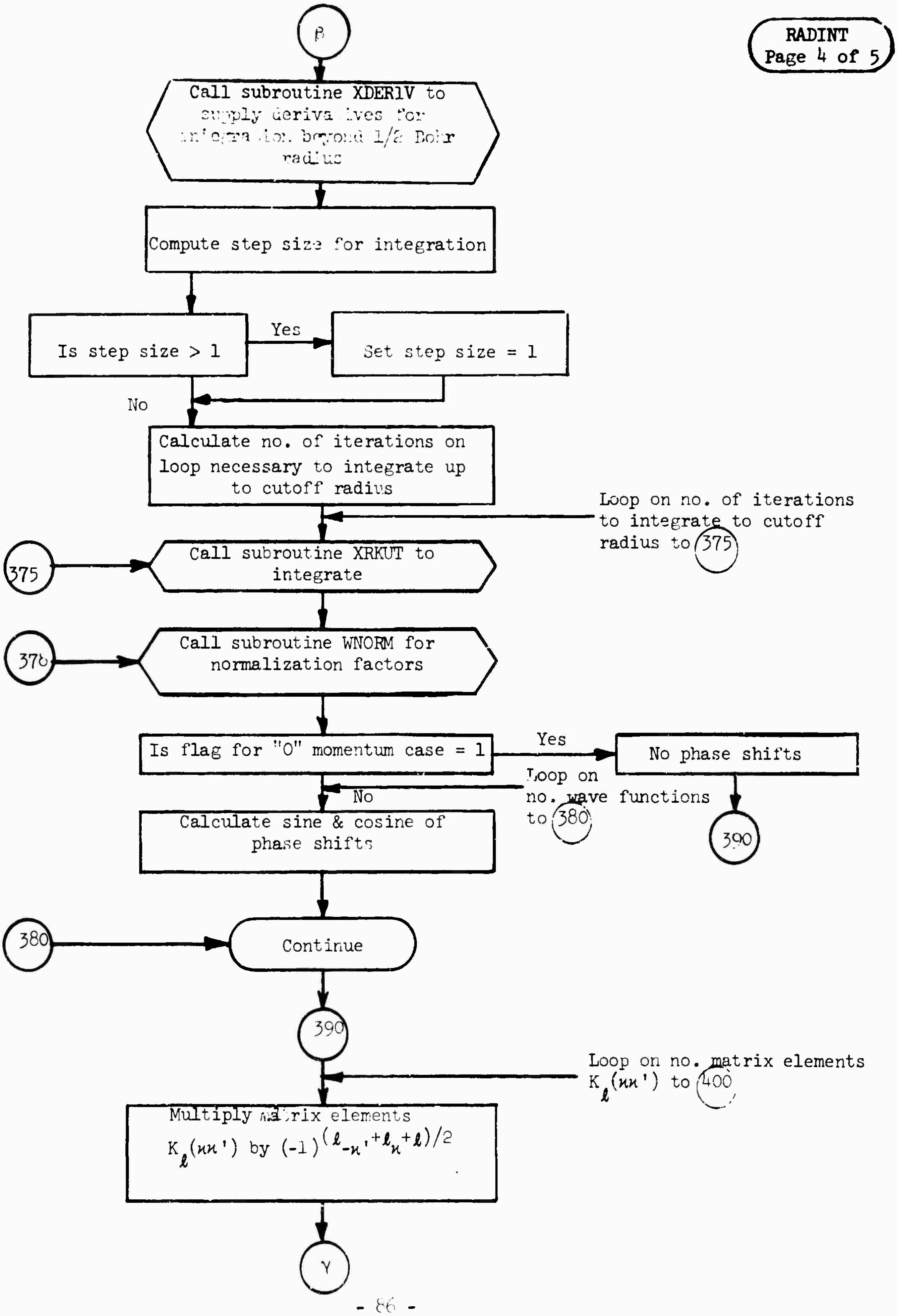
Name	Dimension	Mode	Meaning
E		R	Free electron energy (in mc^2 units)
RK		R	$\sqrt{E^2 - 1}$, free electron momentum
I		I	Loop index of free electron state
TUGAM	30	R	$2\sqrt{\kappa^2 - (\frac{Z}{137.0367})^2} + 1$
IUP		I	Max number of matrix elements
NØ		I	Loop index to initialize to zero components of free electron wavefunction, their derivatives, integrand for matrix elements and their integration storage variables.
N		I	Range on number of iterations for radial integration up to radius of one.
XCUT		R	Radial cutoff on uniform grid in tabulation.
WAVE		R	Max of wave numbers of free electron and photon.

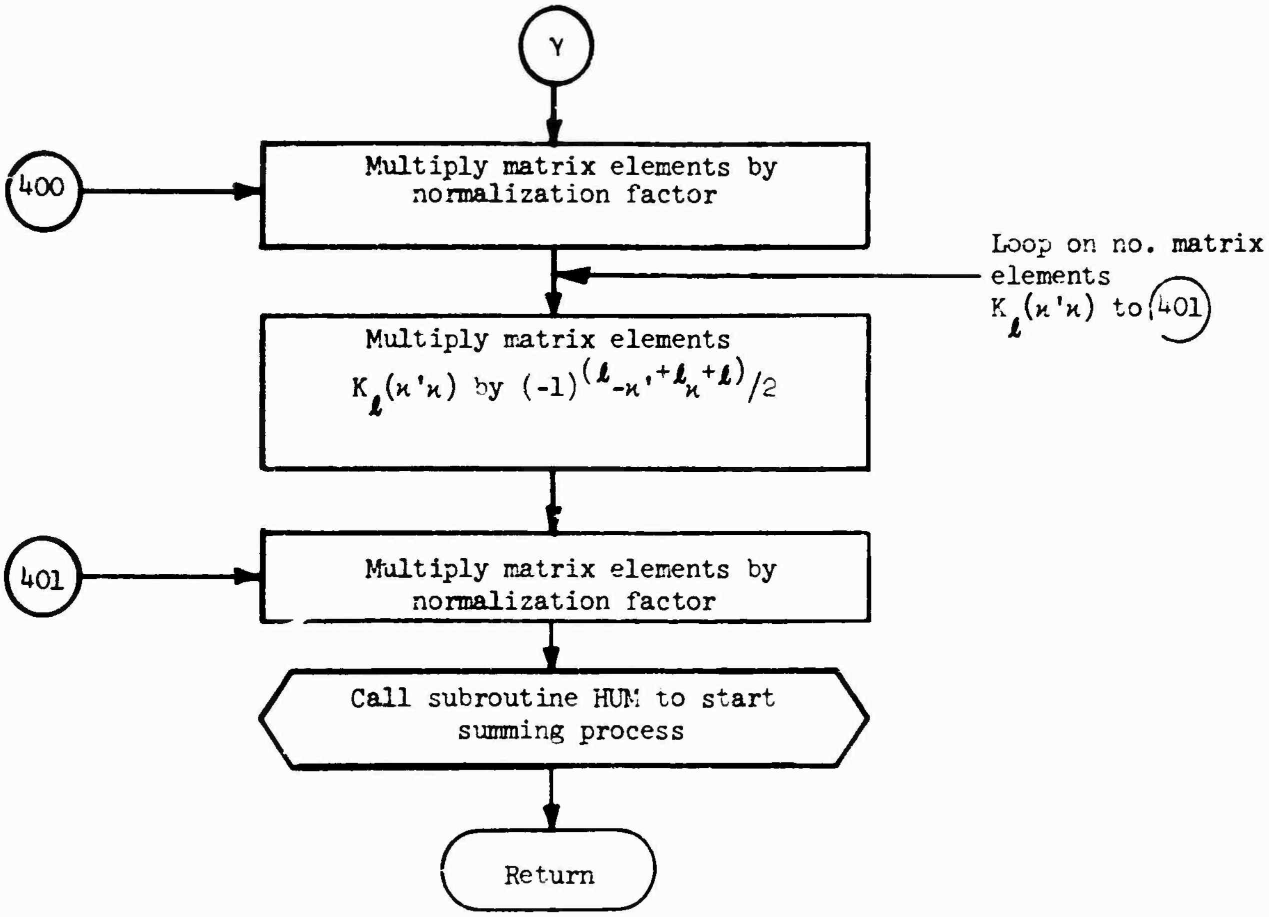
Name	Dimension	Mode	Meaning
LUB		I	Range on number of iterations beyond one-half Bohr radius for radial integration
RX		R	Current radial variable beyond one-half Bohr radius
LTOT		I	$\ell_{-\kappa'} + \ell_{\kappa} + \ell$


RADINT
Page 1 of 5
Enter RADINT
Is K.E. of electron = 0
Yes
Compute asymptotic form for C momentum case
302
No
Set flag for "0" momentum case to zero
Calculate intermediate variables
Loop to on no. wave functions to 300
Compute starting values for integration of wave functions, their derivatives and integration coefficients with pure Coulomb values properly normalized except for one factor
Test ϰ for sign to avoid round-off difficul..es
Is ϰ > 0
Yes
268
No
Compute starting values one way
270
268
Compute starting values alternate way
270
Compute derivatives of wave functions
300
Continue
309


302
Calculate asymptotic form for zero electron momentum
Set flag for 0 momentum case = 1
Loop on no. wave functions to 306
Is ϰ > 0
Yes
304
No
Compute integration coefficients and starting values of wave functions
305
304
Compute integration coefficients and starting values of wave functions alternate way
305
Calculate derivatives of wave functions
306
Continue
309
Set radial index in bound state tabulation to one
Loop on max number of matrix elements to 315
315
Prior to start of integration initialize to 0 matrix elements, their derivatives, and integration storage locations
Loop on free wave functions to 316
316
Initialize to 0 storage locations for wave functions
Set integration step size for integration up to radius of one
α


α
Calculate no. of iterations on loop to integrate up to radius of one
Loop on no. of iterations to integrate up to radius of 1 to 350
350
Call RKUT for integration up to radius = 1
Loop on no. wave functions to 365
365
Insert remaining factor in initial normalization of wave functions, their derivatives & integration storage locations
Loop on no. matrix elements $K_\ell(\varkappa\varkappa')$ to 500
500
Insert factor in matrix elements $K_\ell(\varkappa\varkappa')$, the derivatives and storage locations
Loop on no. matrix elements $K_\ell(\varkappa\,\varkappa)$ to 510
510
Insert factor in matrix elements $K_\ell(\varkappa'\varkappa)$, the derivatives & storage locations
Set step size for integration beyond radius of one, compute radial cutoff on uniform grid tabulation
Call subroutine DERIV to supply derivatives for integration
Calculate no. of iterations on loop necessary to integrate up to radial cutoff
Loop on no. iterations to integrate up to cutoff radius to 370
370
Call RKUT to integrate
Is cutoff radius for integration ≤ radial cutoff on uniform grid
Yes
Integration completed
378
No
Must continue integration by interpolation process
β


β
Call subroutine XDER1V to supply derivatives for integration beyond 1/2 Bohr radius
Compute step size for integration
Is step size > 1
Yes
Set step size = 1
No
Calculate no. of iterations on loop necessary to integrate up to cutoff radius
Loop on no. of iterations to integrate to cutoff radius to 375
375
Call subroutine XRKUT to integrate
376
Call subroutine WNORM for normalization factors
Is flag for "0" momentum case = 1
Yes
No phase shifts
Loop on no. wave functions to 380
No
390
Calculate sine & cosine of phase shifts
380
Continue
390
Loop on no. matrix elements $K_\ell(\kappa\kappa')$ to 400
Multiply matrix elements $K_\ell(\kappa\kappa')$ by $(-1)^{(\ell_{-\kappa'}+\ell_\kappa+\ell)/2}$
γ


Y
400
Multiply matrix elements by normalization factor
Loop on no. matrix elements $K_\ell(\kappa'\kappa)$ to 401
Multiply matrix elements $K_\ell(\kappa'\kappa)$ by $(-1)^{(\ell_{-\kappa'}+\ell_\kappa+\ell)}/2$
401
Multiply matrix elements by normalization factor
Call subroutine HUM to start summing process
Return

```
$IBFTC RADN
      SUBROUTINE RADINT                                                 RAD00010
      COMMON PI,HALFPI,FOURPI,RAD,SQ2,Q,ZA,ZAZA,EFN,EGN,V,CG(30),GAM(30)RAD00020
      COMMON/BESSEL/FL(15),PC(15),OF(15,15),M1,M2,B(15)                 RAD00030
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),HRAD00040
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB           RAD00050
      COMMON/MAT/SF(30),SG(30),FK(200),FKP(200),SFK(200),SFKP(200),RCUT RAD00060
      COMMON /ONWARD/RX,SCX,GBX,FBX                                     RAD00100
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)  RAD00070
      COMMON/VECT/KF(200),KG(200),LBES(200),LBS(200),LKB                RAD00080
      DIMENSION RNORM(30), TUGAM(30)                                    RAD00090
 100  FORMAT(5X,50HLENGTH UNITS ARE HBAR   MC ( 1 BOHR RADIUS = 137 )//)RAD00110
 101  FORMAT (5X,24HINTEGRATION STEP SIZE IS,F11.7, 5X,5HUP TO,F9.3//)  RAD00120
      E = EFN+1.0                                                       RAD00130
      EGN = E+1.0                                                       RAD00140
      IF ( EFN .EQ. 0.)     GO TO 302                                   RAD00150
      RK = SQRT(EFN*EGN)                                                RAD00160
      IEND = 0                                                          RAD00170
      SQEG = SQRT (EGN)                                                 RAD00180
      SQE = RK/SQEG                                                     RAD00190
      RKE = RK / EGN                                                    RAD00200
      GNU = ZA*E/RK                                                     RAD00210
      TRK = 2.0*RK                                                      RAD00220
      TWORK = EXP ( GNU * HALFPI ) / SQRT ( TRK * PI )                  RAD00230
      DO 300 I=1,K2M                                                    RAD00240
      GAM(I) = SQRT (FKAP(I)**2-ZAZA)                                   RAD00250
      TUGAM(I) = 2.0*GAM(I)+1.0                                         RAD00260
      CALL LOGGAM (GAM(I),GNU,XRE,XIM)                                  RAD00270
      ZK = TWORK*(TRK**GAM(I))*EXP(XRE)                                 RAD00280
      IF (FKAP(I).GT.0.0)      GO TO 268                                RAD00290
      CF(I) = FKAP(I)-GAM(I)                                            RAD00300
      CG(I) = -ZAZA/CF(I)                                               RAD00310
      HM = ZAZA * EFN / ( CF(I) * ( E* FKAP(I) - GAM(I) ) )             RAD00320
      HP = 2.0 - HM                                                     RAD00330
      HM = SQRT ( HM )                                                  RAD00340
      HP = SQRT ( HP )                                                  RAD00350
      G(I) = ZK * SQEG * ( GAM(I) * HP + GNU * HM )                     RAD00360
      F(I) = ZA*G(I)/CF(I)                                              RAD00370
      GO TO 270                                                         RAD00380
 268  CG(I) = -FKAP(I)-GAM(I)                                           RAD00390
      CF(I) = -ZAZA/CG(I)                                               RAD00400
      HP = -ZAZA * EGN * ( E*FKAP(I)+GAM(I) ) / CG(I)                   RAD00410
      HP = HP / ( EFN * EGN * FKAP(I) + ZAZA )                          RAD00420
      HM = 2.0 - HP                                                     RAD00430
      HM = SQRT ( HM )                                                  RAD00440
      HP = SQRT ( HP )                                                  RAD00450
      F(I) = ZK * SQE * ( GAM(I) * HM + GNU * HP )                      RAD00460
      G(I) = -ZA*F(I)/CG(I)                                             RAD00470
 270  DG(I) = F(I)*((1.0-CF(I))*EGN-CF(I)*EFN)/TUGAM(I)                 RAD00480
      DF(I) = -G(I)*((1.0-CG(I))*EFN-CG(I)*EGN)/TUGAM(I)                RAD00490
 300  CONTINUE                                                          RAD00500
      GO TO 309                                                         RAD00510
 302  HAZ=SQRT (ZA)                                                     RAD00520
      TUAZ = 2.0*ZA                                                     RAD00530
      IEND = 1                                                          RAD00540
      DO 306 I=1,K2M                                                    RAD00550
      GAM(I) = SQRT (FKAP(I)**2-ZAZA)                                   RAD00560
      TUGAM(I) = 2.0 * GAM(I) + 1.0                                     RAD00570
      F(I) = HAZ*(TUAZ**GAM(I))                                         RAD00580
      IF (FKAP(I).GT.0.0)      GO TO 304                                RAD00590
      F(I) = -F(I)                                                      RAD00600
      CF(I) = FKAP(I)-GAM(I)                                            RAD00610
      CG(I) = -ZAZA/CF(I)                                              RAD00620
      AZKAP = CF(I)/ZA                                                 RAD00630
      GO TO 505                                                        RAD00640
  504 CG(I) = -FKAP(I)-GAM(I)                                          RAD00650
      CF(I) = -ZAZA/CG(I)                                              RAD00660
      AZKAP = -ZA/CG(I)                                                RAD00670
  505 G(I) = AZKAP*F(I)                                                RAD00680
      DF(I) = -F(I)*TUAZ/TUGAM(I)                                      RAD00690
      DG(I) = 2.0*F(I)*(1.0-CF(I))/TUGAM(I)                            RAD00700
  506 CONTINUE                                                         RAD00710
  509 NTAB = 1                                                         RAD00720
      M1 = LM+1                                                        RAD00730
      M2 = LM+2                                                        RAD00740
      IUP = MAX0 (NK,NKP)                                              RAD00750
      DO 515 NO=1,IUP                                                  RAD00760
      FK(NO) = 0.0                                                     RAD00770
      FKP(NO) = 0.0                                                    RAD00780
      DFK(NO) = 0.0                                                    RAD00790
      DFKP(NO) = 0.0                                                   RAD00800
      SFK(NO) = 0.0                                                    RAD00810
  515 SFKP(NO) = 0.0                                                   RAD00820
      DO 516 NO=1,K2M                                                  RAD00830
      SF(NO) = 0.0                                                     RAD00840
  516 SG(NO) = 0.0                                                     RAD00850
      H=0.0078125                                                      RAD00860
      N = 1.0/H + .1                                                   RAD00870
      RONE=1.0                                                         RAD00880
      DO 550 I=1,N                                                     RAD00890
  550 CALL RKUT                                                        RAD00900
      DO 565 I=1,K2M                                                   RAD00910
      CALL LOGGAM (TUGAM(I),0,DUM1,DUM2)                               RAD00920
      RNORM(I) = EXP (-DUM1)                                           RAD00930
      F(I) = F(I)*RNORM(I)                                             RAD00940
      G(I) = G(I)*RNORM(I)                                             RAD00950
      DF(I) = DF(I)*RNORM(I)                                           RAD00960
      DG(I) = DG(I)*RNORM(I)                                           RAD00970
      SF(I) = SF(I)*RNORM(I)                                           RAD00980
      SG(I) = SG(I)*RNORM(I)                                           RAD00990
      CF(I) = FKAP(I)                                                  RAD01000
  565 CG(I) = -FKAP(I)                                                 RAD01010
      DO 500 N=1,NK                                                    RAD01020
      I = KG(N)                                                        RAD01030
      FK(N) = FK(N)*RNORM(I)                                           RAD01040
      DFK(N) = DFK(N)*RNORM(I)                                         RAD01050
  500 SFK(N) = SFK(N)*RNORM(I)                                         RAD01060
      DO 510 N=1,NKP                                                   RAD01070
      I = KF(N)                                                        RAD01080
      FKP(N) = FKP(N)*RNORM(I)                                         RAD01090
      DFKP(N) = DFKP(N)*RNORM(I)                                       RAD01100
  510 SFKP(N) = SFKP(N)*RNORM(I)                                       RAD01110
      WRITE ( 6, 100 )                                                 RAD01120
      WRITE (6,101) H,RONE                                             RAD01130
      H = 0.125                                                        RAD01140
      XCUT = AMIN1 (RCUT,65.0)                                         RAD01150
      WRITE ( 6, 101 ) H, XCUT                                         RAD01160
      NEW = 1                                                          RAD01170
      CALL DERIV                                                       RAD01180
      NDON = (XCUT-1.0)/H+0.1                                          RAD01190
      DO 570 NDO=1,NDON                                                RAD01200
  570 CALL RKUT                                                        RAD01210
      RX = XCUT                                                        RAD01220
      IF (RCUT.LE.XCUT)      GO TO 578                                 RAD01230
      NEW=1                                                             RAD01240
      CALL XDERIV                                                       RAD01250
      WAVE = AMAX1 (RK,Q)                                               RAD01260
      IH=PI/WAVE                                                        RAD01270
      H=FLOAT (IH)/8.0                                                  RAD01280
      IF (H.GT.1.0)     H=1.0                                           RAD01290
      WRITE ( 6, 101 ) H, RCUT                                          RAD01300
      LUB = (RCUT-RX)/H+0.1                                             RAD01310
      DO 375 I=1,LUB                                                    RAD01320
  375 CALL XRKUT                                                        RAD01330
  378 CALL WNORM (RNORM,RX)                                             RAD01340
      IF ( IEND .EQ. 1 )      GO TO 390                                 RAD01350
      M1 = KM + 2                                                       RAD01360
      M2 = KM + 3                                                       RAD01370
      Y = RK * RCUT                                                     RAD01380
      CALL SPHBES(Y)                                                    RAD01390
      DO 380 I=1,K2M                                                    RAD01400
      J1 = LMK(I) + 1                                                   RAD01410
      J2 = LK(I) + 1                                                    RAD01420
      SI(I) = RKE * SN(I) * G(I) * B(J1) - F(I) * B(J2)                 RAD01430
      CR(I) = RKE * G(I) * B(J2) + SN(I) * F(I) * B(J1)                 RAD01440
      RNM = SQRT ( SI(I) * SI(I) + CR(I) * CR(I) )                      RAD01450
      SI(I) = SI(I) / RNM                                               RAD01460
      CR(I) = CR(I) / RNM                                               RAD01470
  380 CONTINUE                                                          RAD01480
  390 DO 400 K=1,NK                                                     RAD01490
      I = KG(K)                                                         RAD01500
      LTOT = LMKB+LK(I)+LBES(K)-1                                       RAD01510
      IF (MOD(LTOT,4).NE.0)     FK(K)=-FK(K)                            RAD01520
  400 FK(K) = FK(K)*RNORM(I)                                            RAD01530
      DO 401 K=1,NKP                                                    RAD01540
      I = KP(K)                                                         RAD01570
      LTOT = LMKB+LK(I)+LBS(K)-1                                        RAD01550
      IF (MOD(LTOT,4).NE.0)     FKP(K)=-FKP(K)                          RAD01560
  401 FKP(K) = FKP(K)*RNORM(I)                                          RAD01580
  420 CALL HUM                                                          RAD01590
      RETURN                                                            RAD01600
      END                                                               RAD01610
```

SUBROUTINE RKUT

Purpose: Performs the Runge-Kutta integration. The routine uses indexed tabulated values of the bound-state wavefunction and the potential obtained previously by interpolation from the Waber output. Radial integration is performed up to a maximum of one-half Bohr radius.

Method: Runge-Kutta Integration (Gill Form). A fourth order integration scheme in which the error in each step is of the order h^5, where h is the interval size.

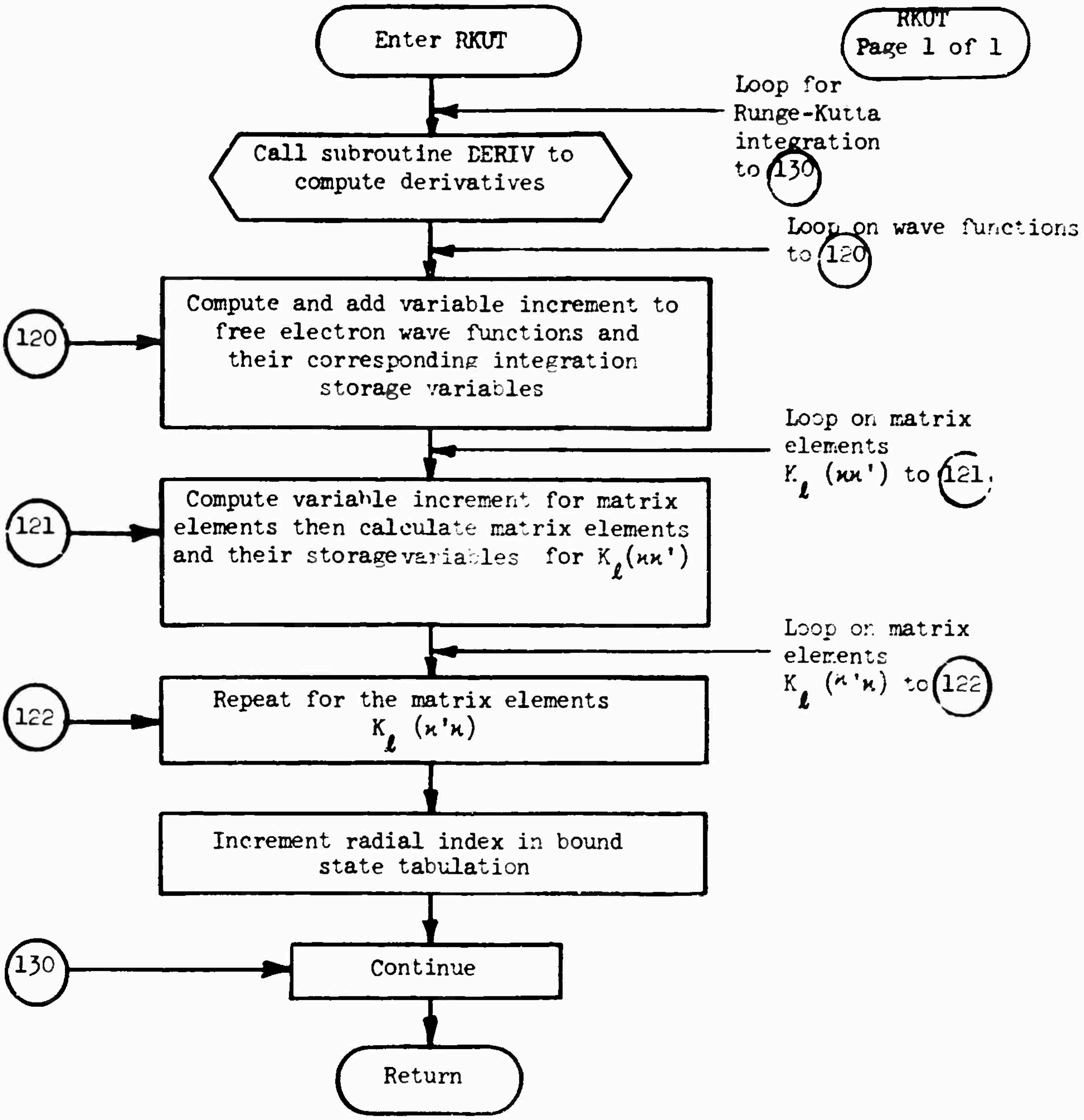
Subroutine called: DERIV

Subroutine called by: RADINT

Labelled Common: DFUNC, KUT, LIMIT, MAT, TAPES

Local Variables:

Name	Dimension	Mode	Meaning
J		I	Loop index on Runge-Kutta integration
I		I	Loop index on; number of free wavefunctions, number of matrix elements $K_\ell(\kappa\kappa')$ and $K_\ell(\kappa'\kappa)$
Z		R	Incremental variable for "small" component of free electron wavefunction
ZP		R	Incremental variable for "large" component of free electron wavefunction


RKUT
Page 1 of 1
Enter RKUT
Loop for Runge-Kutta integration to 130
Call subroutine DERIV to compute derivatives
Loop on wave functions to 120
120
Compute and add variable increment to free electron wave functions and their corresponding integration storage variables
Loop on matrix elements K_ℓ ($\kappa\kappa'$) to 121
121
Compute variable increment for matrix elements then calculate matrix elements and their storage variables for $K_\ell(\kappa\kappa')$
Loop on matrix elements K_ℓ ($\kappa'\kappa$) to 122
122
Repeat for the matrix elements K_ℓ ($\kappa'\kappa$)
Increment radial index in bound state tabulation
130
Continue
Return

```
$IBFTC RKUG
      SUBROUTINE RKUT                                                   RKU00010
C                     RUNGE-KUTTA INTEGRATION                           RKU00020
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),HRKU00030
      COMMON /KUT/RK1(4),RK2(4),RK3(4),RK4(4),K4(4)                     RKU00040
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB           RKU00050
      COMMON/MAT/SF(30),SG(30),FK(200),FKP(200),SFK(200),SFKP(200),RCUT RKU00060
      COMMON /TAPE /X(1500),SCF(1500),FB(1500),GB(1500),GAMB,SCREEN     RKU00070
      DO 130 J=1,4                                                      RKU00080
      CALL DERIV                                                        RKU00090
      DO 120 I=1,K2M                                                    RKU00100
      Z = RK1(J)*(DF(I)-RK2(J)*SF(I))                                   RKU00110
      ZP = RK1(J)*(DG(I)-RK2(J)*SG(I))                                  RKU00120
      F(I) = F(I)+H*Z                                                   RKU00130
      G(I) = G(I)+H*ZP                                                  RKU00140
      SF(I) = SF(I)+3.0*Z-RK3(J)*DF(I)                                  RKU00150
  120 SG(I) = SG(I)+3.0*ZP-RK3(J)*DG(I)                                 RKU00160
      DO 121 I=1,NK                                                     RKU00170
      Z = RK1(J)*(DFK(I)-RK2(J)*SFK(I))                                 RKU00180
      FK(I) = FK(I)+H*Z                                                 RKU00190
  121 SFK(I) = SFK(I)+3.0*Z-RK3(J)*DFK(I)                               RKU00200
      DO 122 I=1,NKP                                                    RKU00210
      Z = RK1(J)*(DFKP(I)-RK2(J)*SFKP(I))                               RKU00220
      FKP(I) = FKP(I)+H*Z                                               RKU00230
  122 SFKP(I) = SFKP(I)+3.0*Z-RK3(J)*DFKP(I)                            RKU00240
      NTAB=NTAB+K4(J)                                                   RKU00250
  130 CONTINUE                                                          RKU00260
      RETURN                                                            RKU00270
      END                                                               RKU00280
```

This page is intentionally left blank.

SUBROUTINE SINDEX

Purpose: To catalogue and index matrix elements consistent with selection rules.

Method: Routine examines the angular momentum values to find and index all the matrix elements compatible with the selection rules and with the input cutoff values of the quantum numbers and records the free electron κ and the photon ℓ values for each matrix element in reference vectors (separately for the two kinds of matrix elements).

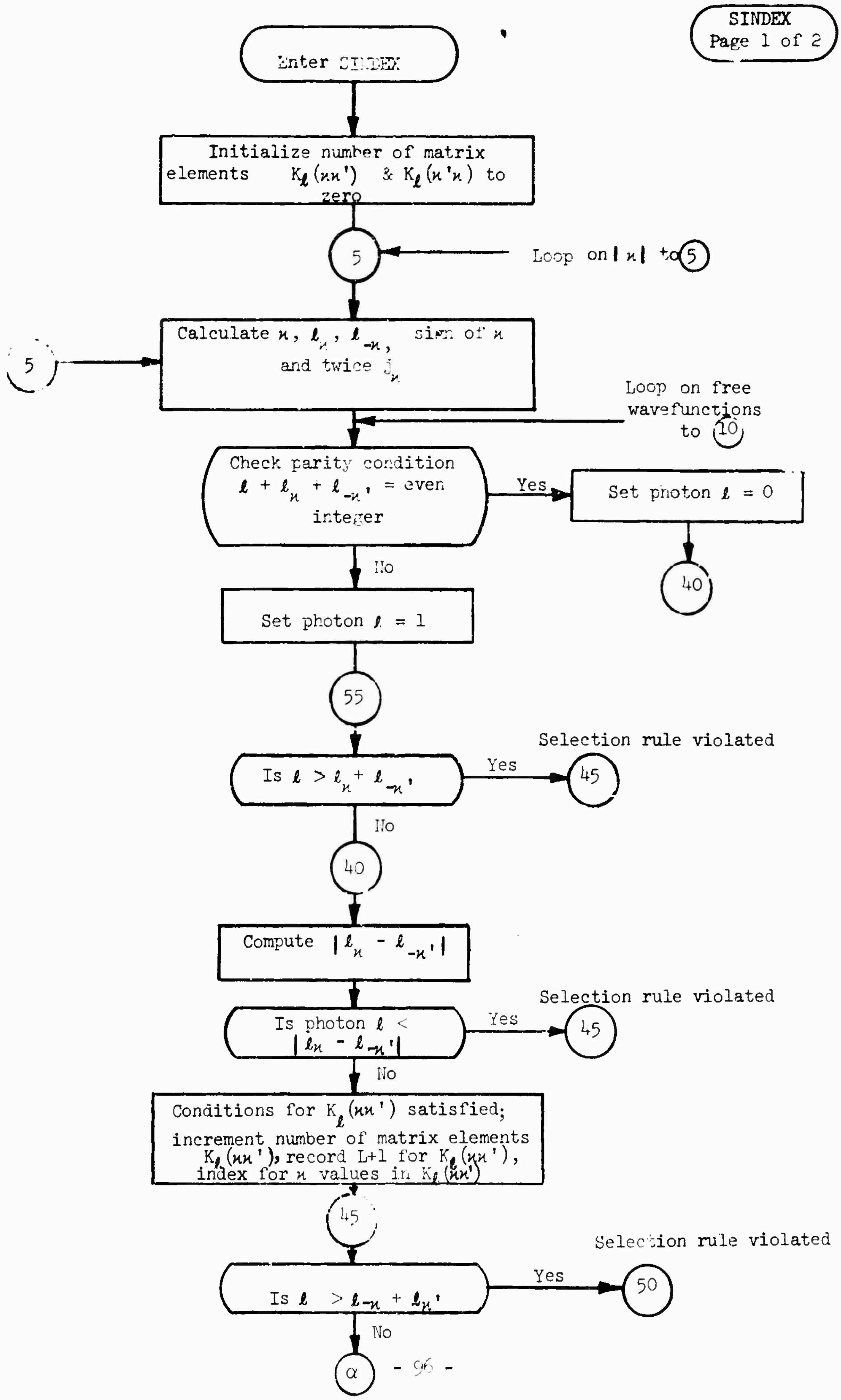
Subroutine called: None

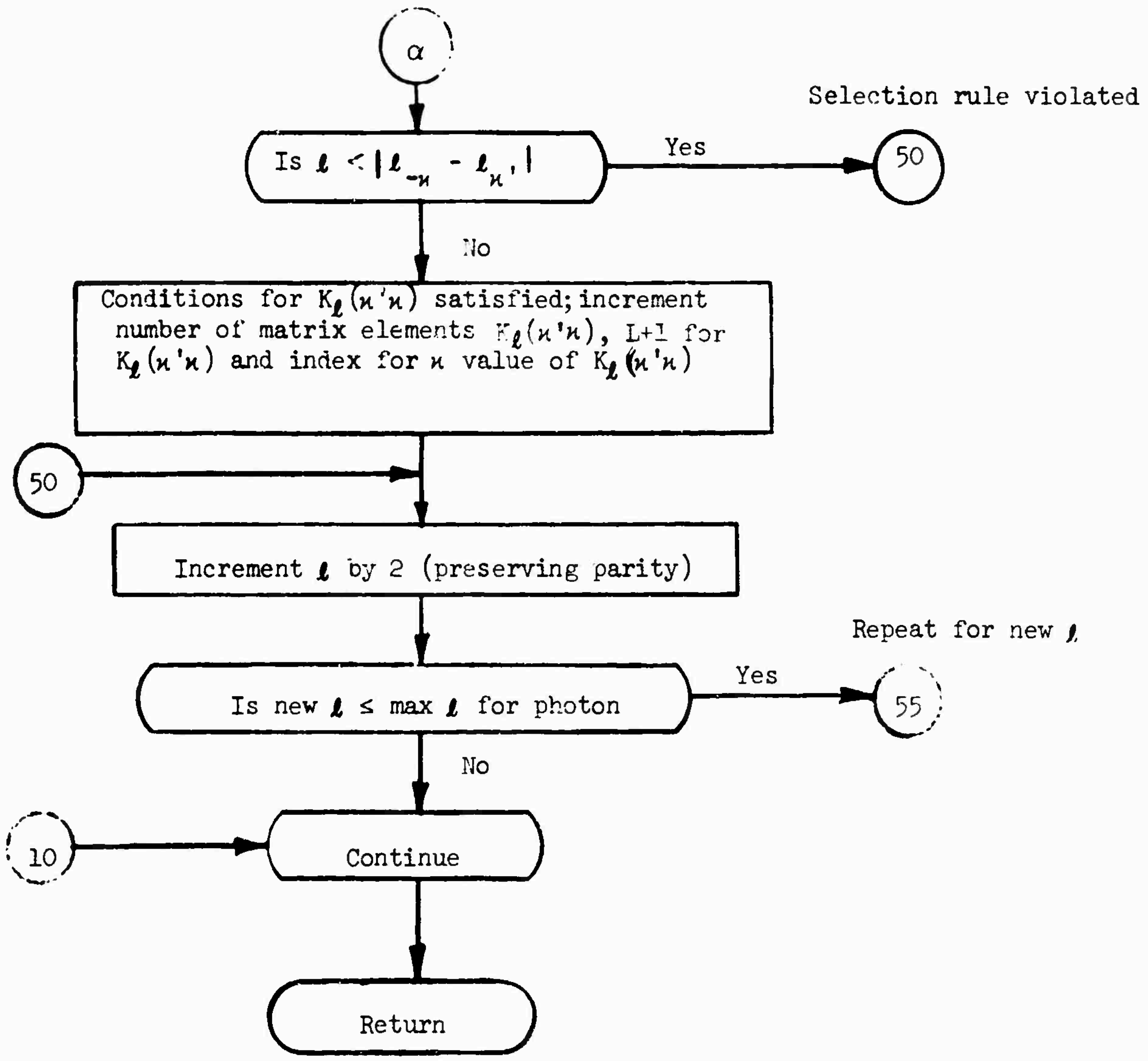
Subroutine called by: PELEC

Labelled Common: QUANT, VECT, LIMIT

Local Variables:

Name	Dimension	Mode	Meaning
I		I	Loop index on max κ for electron
K		I	Loop index on number of free electron states
LSUM		I	$\ell_{\kappa} + \ell_{-\kappa'}$
LDIFF		I	$\lvert \ell_{\kappa} - \ell_{-\kappa'} \rvert$
L		I	Photon angular momentum


SINDEX
Page 1 of 2
Enter SINDEX
Initialize number of matrix elements $K_\ell(\kappa\kappa')$ & $K_\ell(\kappa'\kappa)$ to zero
5
Loop on $|\kappa|$ to 5
5
Calculate κ, ℓ_κ, $\ell_{-\kappa}$, sign of κ and twice j_κ
Loop on free wavefunctions to 10
Check parity condition $\ell + \ell_\kappa + \ell_{-\kappa'}$ = even integer
Yes
Set photon ℓ = 0
40
No
Set photon ℓ = 1
55
Selection rule violated
Is $\ell > \ell_\kappa + \ell_{-\kappa'}$
Yes
45
No
40
Compute $|\ell_\kappa - \ell_{-\kappa'}|$
Selection rule violated
Is photon ℓ < $|\ell_\kappa - \ell_{-\kappa'}|$
Yes
45
No
Conditions for $K_\ell(\kappa\kappa')$ satisfied; increment number of matrix elements $K_\ell(\kappa\kappa')$, record L+1 for $K_\ell(\kappa\kappa')$, index for κ values in $K_\ell(\kappa\kappa')$
45
Selection rule violated
Is $\ell > \ell_{-\kappa} + \ell_{\kappa'}$
Yes
50
No
α


α
Selection rule violated
Is $\ell < |\ell_{-\kappa} - \ell_{\kappa'}|$
Yes
50
No
Conditions for $K_\ell(\kappa'\kappa)$ satisfied; increment number of matrix elements $K_\ell(\kappa'\kappa)$, L+1 for $K_\ell(\kappa'\kappa)$ and index for κ value of $K_\ell(\kappa'\kappa)$
50
Increment ℓ by 2 (preserving parity)
Repeat for new ℓ
Is new $\ell \leq$ max ℓ for photon
Yes
55
No
10
Continue
Return

```
$IBFTC SIND
      SUBROUTINE SINDEX                                                 SND00010
      COMMON/QUANT/LK(30),LMK(30),JK(30),FKAP(30),SN(30),SI(30),CR(30)  SND00020
      COMMON/VECT/KF(200),KG(200),LBES(200),LBS(200),LKB                SND00030
      COMMON /LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB          SND00040
      NK=0                                                              SND00050
      NKP=0                                                             SND00060
      DO 5 I=1,KM                                                       SND00070
      K=2*I-1                                                           SND00080
      J=K+1                                                             SND00090
      FKAP(K)=-I                                                        SND00100
      FKAP(J)=I                                                         SND00110
      LK(K)=I-1                                                         SND00120
      LK(J)=I                                                           SND00130
      LMK(K)=I                                                          SND00140
      LMK(J)=I-1                                                        SND00150
      SN(K)=-1.0                                                        SND00160
      SN(J)=1.0                                                         SND00170
      JK(K)=LK(K)+LMK(K)                                                SND00180
    5 JK(J)=JK(K)                                                       SND00190
      DO 10 K=1,K2M                                                     SND00200
      LSUM = LK(K)+LMKB                                                 SND00210
      IF (MOD(LSUM,2)) 35,30,35                                         SND00220
   30 L=0                                                               SND00230
      GO TO 40                                                          SND00240
   35 L=1                                                               SND00250
   55 IF (L.GT.LSUM)      GO TO 45                                      SND00260
   40 LDIFF = IABS (LK(K)-LMKB)                                         SND00270
      IF (L.LT.LDIFF)      GO TO 45                                     SND00280
      NK=NK+1                                                           SND00290
      LBES(NK)=L+1                                                      SND00300
      KG(NK)=K                                                          SND00310
   45 LT=LMK(K)+LKB                                                     SND00320
      IF (L.GT.LT)      GO TO 50                                        SND00330
      LD=IABS(LMK(K)-LKB)                                               SND00340
      IF (L.LT.LD)      GO TO 50                                        SND00350
      NKP=NKP+1                                                         SND00360
      LBS(NKP)=L+1                                                      SND00370
      KF(NKP)=K                                                         SND00380
   50 L=L+2                                                             SND00390
      IF (L.LE.LM)      GO TO 55                                        SND00400
   10 CONTINUE                                                          SND00410
      RETURN                                                            SND00420
      END                                                               SND00430
```

SUBROUTINE SPHBES

Purpose: Computes the values of the spherical Bessel function.

Method: The zero order function is obtained as sin R/R. For small argument the higher order functions are calculated from the power series expansion. For large argument, the first order function is computed from its explicit sinusoidal representation and the remaining ones obtained by recursion relations. For intermediate arguments, the lower order functions are obtained by recursion, the higher order by power series.

Subroutines called: None

Subroutine called by: DERIV, XDERIV, RADINT

Labelled Common: BESSEL

Argument sequence: (R)

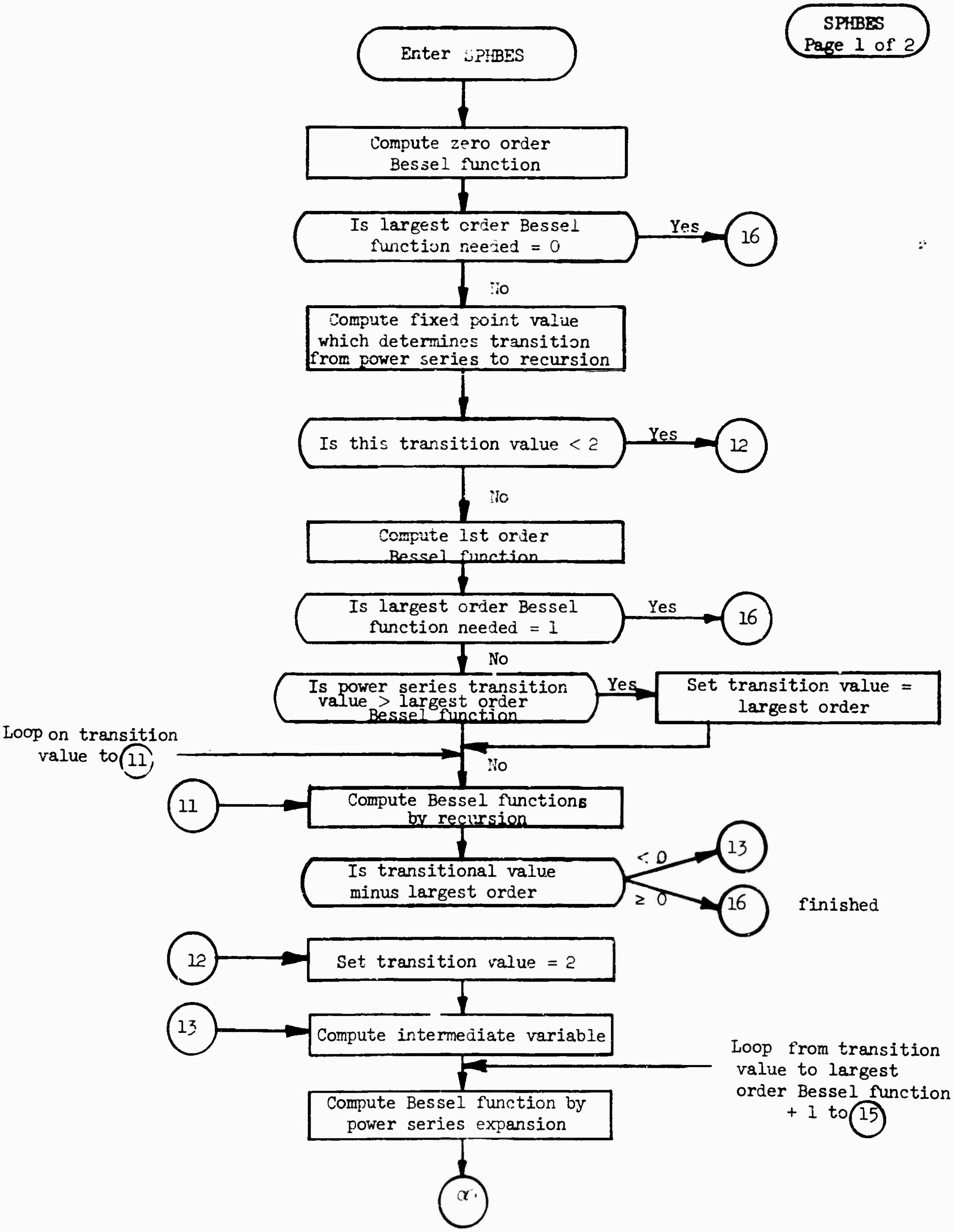
Argument List:

Name	Dimension	Mode	Meaning
R		R	Bessel function argument; photon momentum * radius from DERIV, XDERIV; free electron momentum x radius from RADINT.

Local Variables:

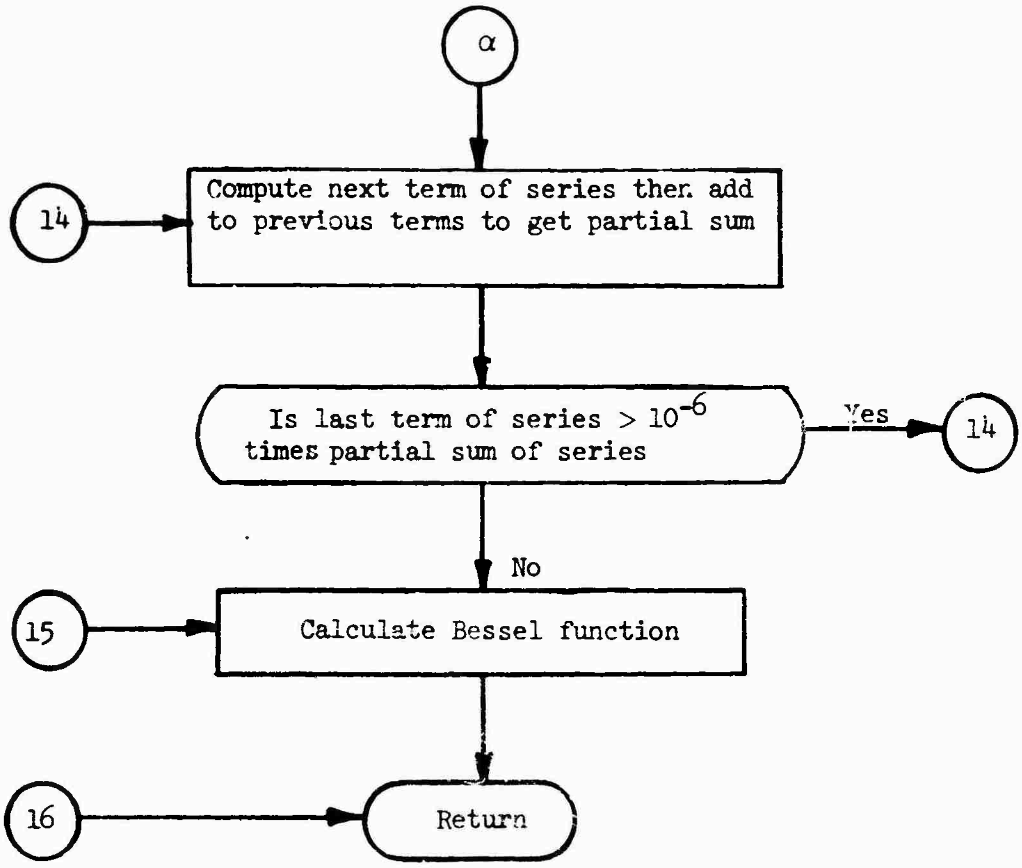
Name	Dimension	Mode	Meaning
NR		I	Defines transition from power series to recursion.
L		I	Loop index on calculating Bessel function.
SER		R	Partial sum of power series expansion.
TER		R	Last term of series expansion.

SPHBES
Page 1 of 2



Enter SPHBES
Compute zero order Bessel function
Is largest order Bessel function needed = 0
Yes
16
No
Compute fixed point value which determines transition from power series to recursion
Is this transition value < 2
Yes
12
No
Compute 1st order Bessel function
Is largest order Bessel function needed = 1
Yes
16
No
Is power series transition value > largest order Bessel function
Yes
Set transition value = largest order
Loop on transition value to 11
No
11
Compute Bessel functions by recursion
Is transitional value minus largest order
< 0
13
≥ 0
16
finished
12
Set transition value = 2
13
Compute intermediate variable
Loop from transition value to largest order Bessel function + 1 to 15
Compute Bessel function by power series expansion
α

SPHBES
Page 2 of 2


α
14
Compute next term of series then add to previous terms to get partial sum
Is last term of series > 10^{-6} times partial sum of series
Yes
14
No
15
Calculate Bessel function
16
Return

```
$IBFTC SPHB
      SUBROUTINE SPHBES (R)
C     SPHERICAL BESSEL FUNCTION
      COMMON/BESSEL/FL(15),PC(15),OF(15,15),M1,M2,B(15)
      OR = 1.0 / R
      B(1) = SIN(R) * OR
      IF ( M1 .EQ. 1 )        GO TO 16
      NR = R + 2.0
      IF ( NR .LT. 2 )        GO TO 12
      B(2) = ( B(1) - COS(R) ) * OR
      IF ( M1 .EQ. 2 )        GO TO 16
      IF ( NR .GT. M1 )       NR=M1
      DO 11 L = 2, NR
   11 B(L+1) = FL(L) * B(L) * OR - B(L-1)
      IF ( NR - M1 )          13,16,16
   12 NR = 2
   13 HAS = 0.5 * R * R
      DO 15 L = NR, M2
      J = 0
      SER = 1.0
      TER = 1.0
   14 J = J + 1
      TER = - TER * HAS * OF(L,J)
      SER = SER + TER
      IF ( ABS(TER) .GT. ( 0.000001 * ABS(SER) ) )             GO TO 14
   15 B(L) = SER * PC(L) * ( R**(L-1) )
   16 RETURN
      END
```

SUBROUTINE WNØRM

Purpose: Computes the normalization factors for the free-electron wavefunction after termination of the numerical integration.

Method: The normalization factors are determined by a matching to the W.K.B. solution. Derivatives of the potential in the W.K.B. expression are obtained from a numerical fit of an exponential to the screening function (ratio of screened to unscreened potential) at the cutoff radius. When the W.K.B. conditions fail (vanishing kinetic energy and high angular momentum), the normalization factor of the previous wavefunctions is used.

Subroutine called: None

Subroutine called by: RADINT

Variables in unlabelled Common: PI, HALFPI, FØURPI, RAD, SQ2, Q, ZA, ZAZA, EFN, EGN, V, CG, GAM

Labelled Common: DFUNC, LIMIT, TAPES

Argument sequence: (RN RM, RX)

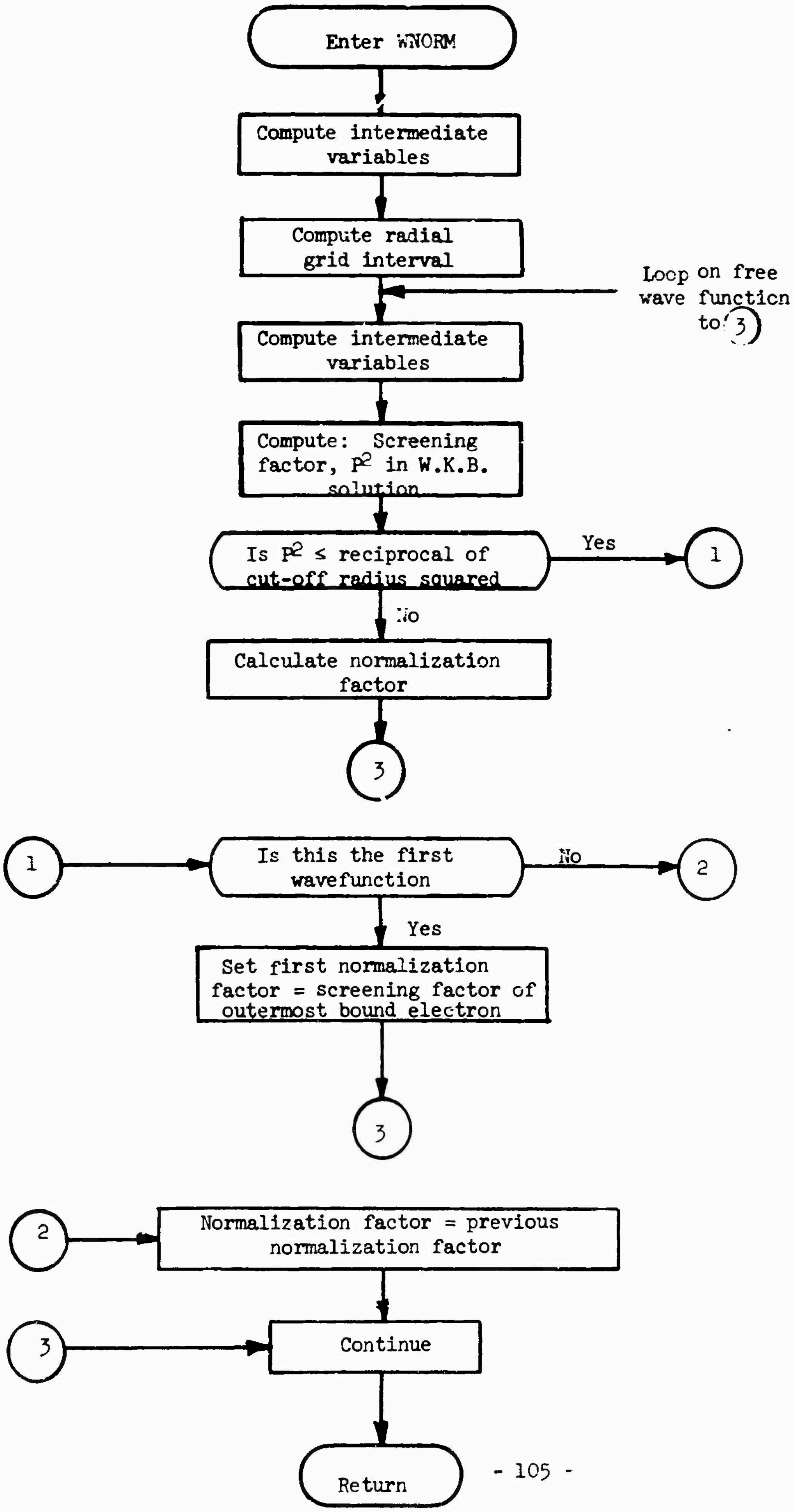
Argument List:

Name	Dimension	Mode	Meaning
RNØRM	30	R	Normalization factors
RX		R	Terminal radius on numerical integration

Local Variables:

Name	Dimension	Mode	Meaning
ØVERR		R	Reciprocial of terminal radius
EMVP1		R	Free electron energy - potential energy + 1
EMV		R	Free electron energy - potential energy
ØRIGH		R	Radial grid interval
I		I	Loop index on number of free electron states

Name	Dimension	Mode	Meaning
RA, RB		R	Tabulated values of screening factor at grid points bracketing terminal radius
EP		R	Screening attenuation coefficient
PSQ		R	P^2 in W.K.B. solution
PSQP		R	Derivative of P^2


Enter WNORM
Compute intermediate variables
Compute radial grid interval
Loop on free wave function to 3
Compute intermediate variables
Compute: Screening factor, P^2 in W.K.B. solution
Is $P^2 \leq$ reciprocal of cut-off radius squared
Yes
1
No
Calculate normalization factor
3
1
Is this the first wavefunction
No
2
Yes
Set first normalization factor = screening factor of outermost bound electron
3
2
Normalization factor = previous normalization factor
3
Continue
Return

```
$IBFTC WNOR
      SUBROUTINE WNORM (RNORM,RX)                                       WNR00010
      COMMON PI,HALFPI,FOURPI,RAD,SQ2,Q,ZA,ZAZA,EFN,EGN,V,CG(30),GAM(30)WNR00020
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),HWNR00030
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB           WNR00040
      COMMON/TAPES/X(1500),SCF(1500),FB(1500),GB(1500),GAMB,SCREEN      WNR00050
      DIMENSION RNORM(30)                                               WNR00060
      OVERR = 1.0 / RX                                                  WNR00070
      OVRRSQ = OVERR*OVERR                                              WNR00080
      EMVP1 = EGN-V                                                     WNR00090
      EMV = EMVP1-1.0                                                   WNR00100
      Y = -V/EMVP1                                                      WNR00110
      ORIGH = X(NTAB+1) - X(NTAB)                                       WNR00120
      DO 3 I=1,K2M                                                      WNR00130
      PK =  CF ( I )                                                    WNR00140
      FKSQ = PK*(PK+1.0)                                                WNR00150
      FKR = PK * OVERR                                                  WNR00160
      FOR = FKSQ * OVRRSQ                                               WNR00170
      RA = SCF(NTAB)                                                    WNR00180
      RB = SCF(NTAB+1)                                                  WNR00190
      EP = ALOG(RA/RB)/ORIGH                                            WNR00200
      V1 = EP + OVERR                                                   WNR00210
      V2 = V1 * V1 + OVRRSQ                                             WNR00220
      V3 = V1 * V2 + 2.0 * OVRRSQ * (V1 + OVERR)                        WNR00230
      V1X = V1*Y                                                        WNR00240
      V2X = V2*Y                                                        WNR00250
      VXSQ = V1X * V1X                                                  WNR00260
      FKVR = V1X*FKR                                                    WNR00270
      PSQ = EMV * EMV - 1.0 + FKVR - 0.75 * VXSQ + 0.5 * V2X - FOR      WNR00280
      IF (PSQ.LE.OVRRSQ)     GO TO 1                                    WNR00290
      PSQP = 2.0 * EMV * V * V1 + (FKR - 1.5*V1X) * VXSQ + 2.0* V1X* V2XWNR00300
     1       - 0.5 * V3 * Y - OVERR * (FKVR + PK*V2X - 2.0*FOR)         WNR00310
      TERM3 = G(I)*(0.5*V1X-FKR+0.25*(PSQP/PSQ))+F(I)*EMVP1             WNR00320
      A = (PI/(SQRT (PSQ)*EMVP1))*(PSQ*G(I) *G(I) +TERM3*TERM3)         WNR00330
      RNORM(I) = 1.0/SQRT (A)                                           WNR00340
      GO TO 3                                                           WNR00350
    1 IF (I.GT.1)     GO TO 2                                           WNR00360
      RNORM(I)=SCREEN                                                   WNR00370
      GO TO 3                                                           WNR00380
    2 RNORM(I)=RNORM(I-1)                                               WNR00390
    3 CONTINUE                                                          WNR00400
      RETURN                                                            WNR00410
      END                                                               WNR00420
```

SUBROUTINE XDERIV

Purpose: Supplies the derivatives for the Runge-Kutta integration of the bound-state wavefunctions and the integrands of the matrix elements for numerical integration beyond one-half Bohr radius.

Method: Calculates the derivatives of the radial components from the coupled Dirac radial equations and the integrand of the matrix elements using values obtained by linear interpolation on the Waber grid of bound state wavefunctions.

Subroutines called: SPHBES, INTERP

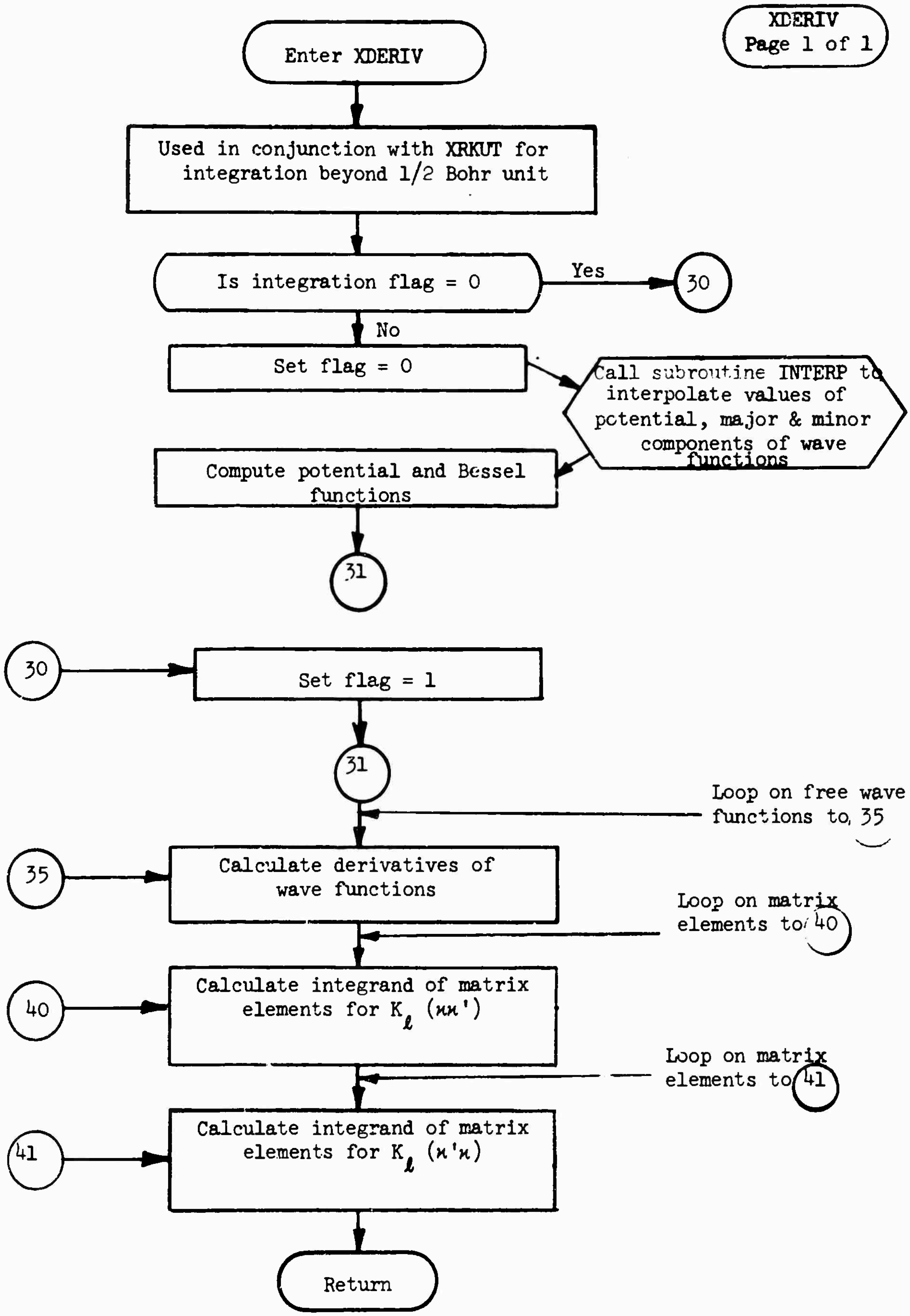
Subroutines called by: XRKUT, RADINT

Labelled Common: BESSEL, DFUNCT, LIMIT, ØNWARD, VECT

Local Variables:

Name	Dimension	Mode	Meaning
Z		R	Photon momentum * radius
N		I	Indexing variable

XDERIV
Page 1 of 1


Enter XDERIV
Used in conjunction with XRKUT for integration beyond 1/2 Bohr unit
Is integration flag = 0
Yes
30
No
Set flag = 0
Call subroutine INTERP to interpolate values of potential, major & minor components of wave functions
Compute potential and Bessel functions
31
30
Set flag = 1
31
Loop on free wave functions to 35
35
Calculate derivatives of wave functions
Loop on matrix elements to 40
40
Calculate integrand of matrix elements for K_ℓ $(\kappa\kappa')$
Loop on matrix elements to 41
41
Calculate integrand of matrix elements for K_ℓ $(\kappa'\kappa)$
Return

```
$IBFTC XDER
      SUBROUTINE XDERIV                                                 XDR00010
C                            COMPUTES DERIVATIVES                       XDR00020
      COMMON PI,HALFPI,FOURPI,RAD,SQ2,Q,ZA,ZAZA,EFN,EGN,V,CG(30),GAM(30)XDR00030
      COMMON/BESSEL/FL(15),PC(15),OF(15,15),M1,M2,B(15)                 XDR00040
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),HXDR00050
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB           XDR00060
      COMMON /ONWARD/RX,SCX,GBX,FBX                                     XDR00070
      COMMON/VECT/KF(200),KG(200),LBES(200),LBS(200),LKB                XDR00080
      IF (NEW.EQ.0)      GO TO 30                                       XDR00090
      NEW = 0                                                           XDR00100
      CALL INTERP                                                       XDR00110
      V = - SCX / RX                                                    XDR00120
      Z=Q*RX                                                            XDR00130
      CALL SPHBES (Z)                                                   XDR00140
      GO TO 31                                                          XDR00150
   30 NEW = 1                                                           XDR00160
   31 DO 35 N=1,K2M                                                     XDR00170
      DF(N) = CF(N)*F(N)/RX-(EFN-V)*G(N)                                XDR00180
   35 DG(N) = CG(N)*G(N)/RX+(EGN-V)*F(N)                                XDR00190
      DO 40 N=1,NK                                                      XDR00200
      I = KG(N)                                                         XDR00210
      L = LBES(N)                                                       XDR00220
   40 DFK(N) = B(L)*G(I)*FBX                                            XDR00230
      DO 41 N=1,NKP                                                     XDR00240
      I = KF(N)                                                         XDR00250
      L = LBS(N)                                                        XDR00260
   41 DFKP(N) = B(L)*F(I)*GBX                                           XDR00270
      RETURN                                                            XDR00280
      END                                                               XDR00290
```

This page is intentionally left blank.

SUBROUTINE XRKUT

Purpose: Performs the Runge-Kutta integration for radial values beyond one-half Bohr radius. The routine uses linearly interpolated values of the bound-state wavefunctions and the potential.

Method: Runge-Kutta Integration (Gill Form)

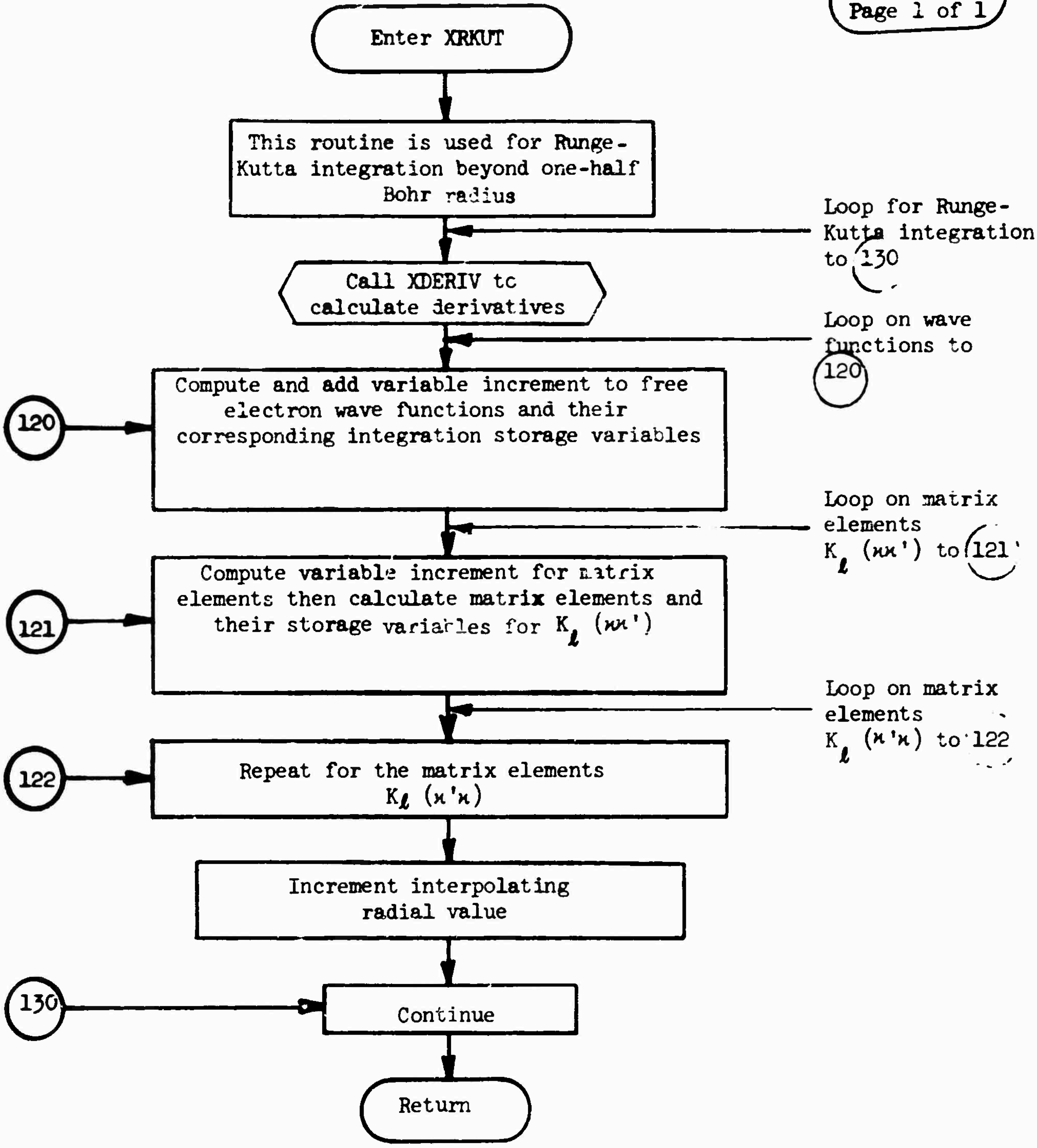
Subroutine called: XDERIV

Subroutine called by: RADINT

Labelled Common: DFUNC, KUT, LIMIT, MAT ØNWARD

Local Variables:

Name	Dimension	Mode	Meaning
J		I	Loop index on Runge-Kutta integration
I		I	Loop index on; number of free wavefunctions, number of matrix elements $K_{\ell}(\kappa\kappa')$ and $K_{\ell}(\kappa'\kappa)$
Z		R	Incremental variable for "small" component of free electron wavefunction
ZP		R	Incremental variable for "large" component of free electron wavefunction


Enter XRKUT
This routine is used for Runge-Kutta integration beyond one-half Bohr radius
Loop for Runge-Kutta integration to 130
Call XDERIV to calculate derivatives
Loop on wave functions to 120
120
Compute and add variable increment to free electron wave functions and their corresponding integration storage variables
Loop on matrix elements K_ℓ $(\kappa\kappa')$ to 121
121
Compute variable increment for matrix elements then calculate matrix elements and their storage variables for K_ℓ $(\kappa\kappa')$
Loop on matrix elements K_ℓ $(\kappa'\kappa)$ to 122
122
Repeat for the matrix elements K_ℓ $(\kappa'\kappa)$
Increment interpolating radial value
130
Continue
Return

```
$IBFTC XRKU
      SUBROUTINE XRKUT
C                     RUNGE-KUTTA INTEGRATION
      COMMON/DFUNC/F(30),G(30),DF(30),DG(30),DFK(200),DFKP(200),CF(30),H
      COMMON /KUT/ RK1(4),RK2(4),RK3(4),RK4(4),K4(4)
      COMMON/LIMIT/JM,LM,KM,K2M,IEND,NEW,NK,NKP,JKB,LMKB,NTAB
      COMMON/MAT/SF(30),SG(30),FK(200),FKP(200),SFK(200),SFKP(200),RCUT
      COMMON /ONWARD/RX,SCX,GBX,FBX
      DO 130 J=1,4
      CALL XDERIV
      DO 120 I=1,K2M
      Z = RK1(J)*(DF(I)-RK2(J)*SF(I))
      ZP = RK1(J)*(DG(I)-RK2(J)*SG(I))
      F(I) = F(I)+H*Z
      G(I) = G(I)+H*ZP
      SF(I) = SF(I)+3.0*Z-RK3(J)*DF(I)
  120 SG(I) = SG(I)+3.0*ZP-RK3(J)*DG(I)
      DO 121 I=1,NK
      Z = RK1(J)*(DFK(I)-RK2(J)*SFK(I))
      FK(I) = FK(I)+H*Z
  121 SFK(I) = SFK(I)+3.0*Z-RK3(J)*DFK(I)
      DO 122 I=1,NKP
      Z = RK1(J)*(DFKP(I)-RK2(J)*SFKP(I))
      FKP(I) = FKP(I)+H*Z
  122 SFKP(I) = SFKP(I)+3.0*Z-RK3(J)*DFKP(I)
      RX = RX+RK4(J)*H
  130 CONTINUE
      RETURN
      END
```

This page is intentionally left blank.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Union Carbide Corporation

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Low Energy Photoelectric Cross Section Calculations
Vol. I: Theory and Results; Vol. II: Program Description

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report - Inclusive dates of research were 1 June 1966 to 30 June 1967

5. AUTHOR(S) *(Last name, first name, initial)*

Vol. I - Brysk, Henry; Zerby, Clayton D.
Vol. II - Brysk, Henry; Glick, Arnold

6. REPORT DATE 31 July 1967	7a. TOTAL NO OF PAGES I - 54; II - 120	7b. NO OF REFS I - 18; II - none

8a. CONTRACT OR GRANT NO.

DA-49-146-XZ-511

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

UCC/DSSD - 299

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

DASA-2023, Volume I and DASA-2023-1, Vol II

10. AVAILABILITY/LIMITATION NOTICES

Not for open publication.
Each transmittal of this document outside the agencies of the U.S. Government must have prior approval of the Director, DASA, Washington, D. C. 20305.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Air Force Weapons Laboratory

13. ABSTRACT